

Predict®

Predefined Object Types

Manual Order Number: PRD411-032ALL

This document applies to the Predict software package at Version 4.1 and to all subsequent versions, unless otherwise indicated in new editions.

Specifications contained herein are subject to change, and these changes will be reported in subsequent release notes or new editions.

Readers' comments are welcomed. Comments may be addressed to the Documentation Department at the address on the back cover. Internet users may send comments to sagdoc@sagus.com.

Requests for publications should be addressed to Software AG or your nearest Software AG affiliate.

Copyright (c) December 1999 by Software AG

Software AG, Adabas, Adaplex+, Com-plete, Elite, Entire, Entire Access, Entire Net-work, Natural, Natural Architect, Natural Connection, Natural Construct, Natural Elite, Natural Expert, Natural New Dimension, Natural Vision, New Dimension, PAC, Predict, SourcePoint, Super Natural, and Tapestry are registered trademarks of Software AG.

Software AG and/or all Software AG products are either trademarks or registered trademarks of Software AG. Other products and company names mentioned herein may be the trademarks of their respective owners.

TABLE OF CONTENTS

PREFACE	1
 1. GENERAL INFORMATION	 5
The Predict Metastructure	6
Global Attributes	7
Object ID	7
Copy ID	9
In Object	9
Restrictions	9
Keys	10
Abstract	10
EDIT Line Options	11
 2. DATABASE	 13
The Database Maintenance Menu	15
Defining Basic Attributes of Databases	17
Documenting Databases of Different Types	19
Database Type A (Adabas C)	19
Specifying the Size of an Adabas C Database	21
Database Types C, E, P (Conceptual, General SQL Handler, Entire System Server Nodes) ..	23
Database Type D (DB2)	24
Database Types Q, M, R, H (Adabas SQL Handler, RMS Handler, rdb Handler, Other Handler)	26
Database Type I (IMS)	27
Database Type T (Target Node)	28
Database Type V (VSAM Handler)	29
Other SQL Database Types	30
Database-Specific Maintenance	31
Purge Database (Code P)	31
Rename/Renumber/Retype Database (Code N)	33
Special Functions for Editing the File List of a Database	38

Predefined Object Types in Predict

Database Retrieval	39
Database-Specific Retrieval Parameters	39
Database-Specific Retrieval Functions	40
Layout of Database Lists	41
Output Options for Database Retrieval	42
3. DATASPACE	47
The Dataspace Maintenance Menu	48
The Add Dataspace Screen	49
The Add/Copy/Modify Dataspace Screen (DB2)	50
The Add/Copy/Modify Dataspace Screen (SQL/DS)	56
Dataspace-Specific Maintenance	57
Purge Dataspace (Code P)	57
Dataspace Retrieval	58
Layout of Dataspace Lists	58
Output Options for Dataspace Retrieval	59
4. EXTRACT	63
The Extract Maintenance Menu	65
The Add/Copy/Modify Extract Screen	66
Extract-Specific Maintenance Functions	67
Copy Extracts (Code C)	67
Operate on Extracts (Code T)	67
Export an Extract (Code U)	72
Edit/Link Objects (Code E)	72
Extract Object Editor	72
Link Objects to Extract (Code E)	78
Build/Extend an Extract (Code B)	81
Purge Extract (Code P)	84
Extract Retrieval	85
Extract-specific Retrieval Functions	85
Output Options for Extract Retrieval	86

5. FIELD	91
The Field Maintenance Menu	94
Defining Basic Attributes of Fields	96
Field Type	99
Level Number	100
Field Format	101
Character Set (SQL)	102
Field Length	103
Descriptor Type	114
Descriptor Type (continued)	115
Maximum Number of Values / Occurrences	116
Unique Option	116
Field Short Name	117
Suppression / Null Value Option	118
Variable Length Option (IMS)	119
Null Default Option	119
Natural Field Length	120
Related Standard File	120
Check against standard	120
Natural Attributes	121
EDIT Line Options	122
Defining Derived Fields	123
General Rules for Defining Derived Fields	124
Defining Derived Fields of Special Types	126
Rules Applying to Format Changes	129
Validation of Derived Field Definitions	133
Defining More Attributes of Fields	140
3GL Specification	141
Condition Name & Value	143
Field Name Synonyms	145
Old Mode Synonyms	146
Adabas Security & Edit mask	147
Field Procedure	148
Derived Field Expression	149
Index Definition (DB2)	150
Default name	154
Constraint name	155

Predefined Object Types in Predict

Field Maintenance	156
Add a Field (Code A)	157
Copy Field (Code C)	157
Move Field within a File (Code H)	158
Purge Field (Code P)	158
Redefine Field (Code R)	159
Browse Through Fields of a File (Code B)	164
Link Verification (Code L)	164
Edit Field Expression (Code Y)	164
Field Retrieval	165
Field-Specific Retrieval Parameters	165
Sorting Fields and Files	167
Field-specific Retrieval Functions	169
Layout of Field Lists	171
Output Options for Field Retrieval	173
 6. FILE	177
The File Maintenance Menu	180
Common File Attributes	183
Defining Basic File Attributes	186
Documenting Files of Different Types	188
Adabas C Files, File Type A	188
File Types Conceptual, Standard and Other	200
SQL File Types	201
Naming Conventions for SQL Objects	202
Common Parameters for SQL File Types	204
Field Lists of SQL Views	206
Editing the Subquery of an SQL View	209
Documenting SQL Tables and Views of Different Types	211
Adabas SQL Server	212
Overview	212
Naming Conventions	212
Adabas Cluster Table	214
Adabas C SQL View	216

Adabas D	217
Naming Conventions	217
Adabas D Table, File Type BT	218
Adabas D View, File Type BV	219
DB2	220
Naming Conventions	220
DB2 Table, File Type D	221
DB2 View, File Type E	223
INFORMIX	224
Naming Conventions	224
INFORMIX Table, File Type XT	225
INFORMIX View, File Type XV	227
INGRES	228
Naming Conventions	228
INGRES Table, File Type JT	229
INGRES View, File Type JV	231
ORACLE	232
Naming Conventions	232
ORACLE Table, File Type OT	233
ORACLE View, File Type OV	236
SYBASE	237
Naming Conventions	237
SYBASE Table, File Type YT	238
SYBASE View, File Type YV	239
General SQL File, File Type X	240
rdb	242
IMS	243
VSAM	247
ISAM	251
Entire System Server	253

Predefined Object Types in Predict

File-Specific Maintenance	255
Purge File (Code P)	256
Rename File (Code N)	257
Edit List of Fields (Code L)	258
Force Standard (Code F)	261
Push Backward (Code B)	261
Modify Adabas Attributes (Code J)	264
Modify STARTAB Elements (Code K)	264
Edit Subquery of a File (Code Y)	264
Rippling – Ensuring Consistent Data Definitions	265
Overview	266
Rippling from Standard Files	269
Rippling from Master Files to Views/Usersviews	273
File Retrieval	279
The File Retrieval Screen	280
File-specific Retrieval Parameters	280
File-specific Retrieval Functions	281
Layout of File Lists	284
Output Options for File Retrieval	286
7. FILE RELATION	291
File Relation Maintenance	292
The File Relation Maintenance Menu	292
The Add a File Relation Screen	293
Validity Checks for File Relations	296
File Relation Retrieval	298
File Relation Specific Retrieval Parameter	298
Layout of File Relation Lists	298
Output Options for File Relation Retrieval	298
8. INTERFACE	299
The Interface Maintenance Menu	300
The Add an Interface Screen	301
Interface Retrieval	302
Output Options for Interface Retrieval	302

9. KEYWORD	303
Keyword Maintenance	304
The Add/Copy/Modify Keyword Screen	305
Keyword Maintenance Functions	305
Keyword Retrieval	309
Keyword-specific Retrieval functions	309
Layout of Keyword Lists	309
Output Options for Keyword Retrieval	310
 10. LIBRARY STRUCTURE	 315
The Library Structure Maintenance Menu	316
The Add/Copy/Modify Library Structure Screen	317
Library Structure Maintenance	318
Function Link Children (Code L)	318
Library Structure Retrieval	318
Output Options for Library Structure Retrieval	318
 11. METHOD	 319
The Method Maintenance Menu	320
The Add a Method Screen	321
Method Retrieval	322
Output Options for Method Retrieval	322
 12. NETWORK	 323
The Network Maintenance Menu	324
The Add a Network Screen	325
Network-Specific Maintenance	325
Purge Network (Code P)	325
Network Retrieval	326
ADASTAR Numbers (Code N)	326
Sample Output	326
Layout of Network Lists	327
Output Options for Network Retrieval	327

13. NODE	329
The Node Maintenance Menu	330
The Add a Node Screen	331
Node Retrieval	332
Output Options for Node Retrieval	332
14. PACKAGELIST	333
The Packagelist Maintenance Menu	334
The Add a Packagelist Screen	336
Packagelist Specific Maintenance	338
Purge Packagelist (Code P)	338
Packagelist Retrieval	339
Packagelist-specific Retrieval Parameter	339
Layout of Packagelist Lists	339
Output Options for Packagelist Retrieval	339
15. PROGRAM	341
The Program Maintenance Menu	343
The Add a Program Screen	347
Defining More Attributes of Programs	349
Programs of Type Class	350
Programs of Type Resource	351
System Programs	352
Programs of Type dynamic	352
Program Maintenance	353
Editing Child Lists	353
Overview of Language-Specific Program Types	354
Generating Database Request Modules (DBRMs) from Program Objects of Language Q (Static SQL)	357
Purge Program (Code P)	358
Redocument Program (Code X)	358
Edit procedure code of a program (Code Y)	366

Program Retrieval	367
Program-Specific Retrieval Parameters	367
Function Programs with Children with Child Type Program	367
Layout of Program Lists	368
Output Options for Program Retrieval	369
16. PROPERTY	373
The Property Maintenance Menu	374
The Add a Property Screen	375
Property Retrieval	376
Output Options for Property Retrieval	376
17. REPORT LISTING	377
The Report Listing Maintenance Menu	378
Modify Report Listing Screen	380
Report Listing Retrieval	381
Layout of Report Listing Lists	381
Output Options for Report Listing Retrieval	381
18. SERVER	383
The Server Maintenance Menu	384
The Add a Server Screen	385
Server Retrieval	386
Layout of Server Lists	386
Output Options for Server Retrieval	386
19. STORAGE SPACE	387
The Storage space Maintenance Menu	388
The Add a Storage space Screen	389
Storage space-Specific Maintenance	390
Purge Storage space (Code P)	390
Storage space Retrieval	391
Unused Storage spaces (Code N)	391
Layout of Storage space Lists	391
Output Options for Storage space Retrieval	392

20. SYSTEM	397
The System Maintenance Menu	398
The Add/Copy/Modify System Screen	400
System-Specific Maintenance	401
Identifying Systems	401
Purge System (Code P)	401
Rename System (Code N)	401
System Retrieval	402
System-Specific Retrieval Parameter	402
Systems with Children (Code T), with Child Type Program	402
Layout of System Lists	402
Output Options for System Retrieval	403
21. USER/OWNER	405
The User/Owner Maintenance Menu	407
The Add a User Screen	409
User Maintenance	411
Purge User	411
User Retrieval	412
User-Specific Retrieval Parameter	412
User-Specific Retrieval Functions	412
Layout of User Lists	412
Output Options for User Retrieval	413
Owner Maintenance	417
Owner-specific Maintenance Functions	418
Owner Retrieval	420
Owner-specific Retrieval Functions	420
Layout of Owner Lists	422
Layout of Owner Lists for Users	423
Output Options for Owner Retrieval	424

22. VERIFICATION 427

 The Verification Maintenance Menu 429

 The Add a Verification Screen 431

 Verification-Specific Maintenance 435

 Purge Verification (Code P) 435

 Rename/Change Status of a Verification (Code N) 435

 Edit Rule of a Verification (Code R) 435

 The Rule Editor 436

 Verification-Specific Retrieval 438

 Verification-Specific Retrieval Parameters 438

 Verification Specific Retrieval Functions 438

 Layout of Verification Lists 439

 Output Options for Verification Retrieval 440

23. VIRTUAL MACHINE 445

 The Virtual Machine Maintenance Menu 446

 The Add a Virtual Machine Screen 447

 Virtual Machine Retrieval 448

 Virtual Machine Specific Retrieval Parameters 448

 Layout of Virtual Machine Lists 448

 Output Options for Virtual Machine Retrieval 448

INDEX 449

PREFACE

This manual describes all the predefined object types in Predict. Only type-specific attributes of the respective object type and the type-specific maintenance and retrieval functions are explained.

Each object type is described in a separate chapter. The object types are arranged in alphabetical order.

How this Manual is Organized

Chapter 1: General Information page 5

The first chapter of this manual provides general information on the predefined object types in Predict. It describes global attributes such as object ID, keywords and restrictions. This general information is not repeated in the descriptions of the individual objects.

Chapter 2: Database page 12

Objects of type *database* document a collection of physical and/or logical files.

Chapter 3: Dataspace page 46

Objects of type *dataspace* document DB2 tablespaces and SQL/DS DBspaces.

Chapter 4: Extract page 62

With this object type you can create sets of objects. An extract is used primarily for transferring data with the Predict Coordinator.

Chapter 5: Field page 90

With the object type *field* you can document field definitions for a wide range of database management systems.

Chapter 6: File page 176

With the object type *file* you can document file structures for a wide range of database management systems. This chapter also describes the process of rippling.

Chapter 7: File Relation page 290

This object type was called *relationship* in earlier versions of Predict. The name was changed for reasons of compatibility with other Software AG products.

With file relations you can document the relationships between fields in a file.

Predefined Object Types in Predict

Chapter 8: Interface, page 298

Together with objects of type *method* and *program*, interfaces document the Natural program object class.

Chapter 9: Keyword page 302

You can assign objects of type *keyword* to other objects in order to link objects logically.

Chapter 10: Library Structure page 314

This object type supports the Steplib concept in Natural.

Chapter 11: Method, page 318

This object type documents the methods of an interface.

Chapter 12: Network page 322

Together with objects of type *virtual machine*, networks document the hardware and operating system environment of a data processing system.

Chapter 13: Node page 328

This object type was introduced together with object type *server* to document Remote Procedure Calls.

Chapter 14: Packagelist page 332

This object type documents DB2 packages.

Chapter 15: Program page 340

With objects of type *program* you can document nearly 20 types of program. Around a dozen different programming languages are supported.

Chapter 16: Property, page 372

This object type documents the properties of an interface.

Chapter 17: Report Listing page 376

This object type was introduced to log transfer operations with the Coordinator and conversion functions.

Chapter 18: Server page 382

This object type is used together with object type *node* to document Remote Procedure Calls.

Chapter 19: Storagespace page 386

This object type documents DB2 storagegroups.

Chapter 20: System page 396

With this object type you can document complex applications.

Chapter 21: User/Owner page 404

An object of type *user* documents an individual user. Several users can be assigned to an *owner* to represent organizational units. These owners can be assigned to other objects to link objects logically.

Chapter 22: Verification page 426

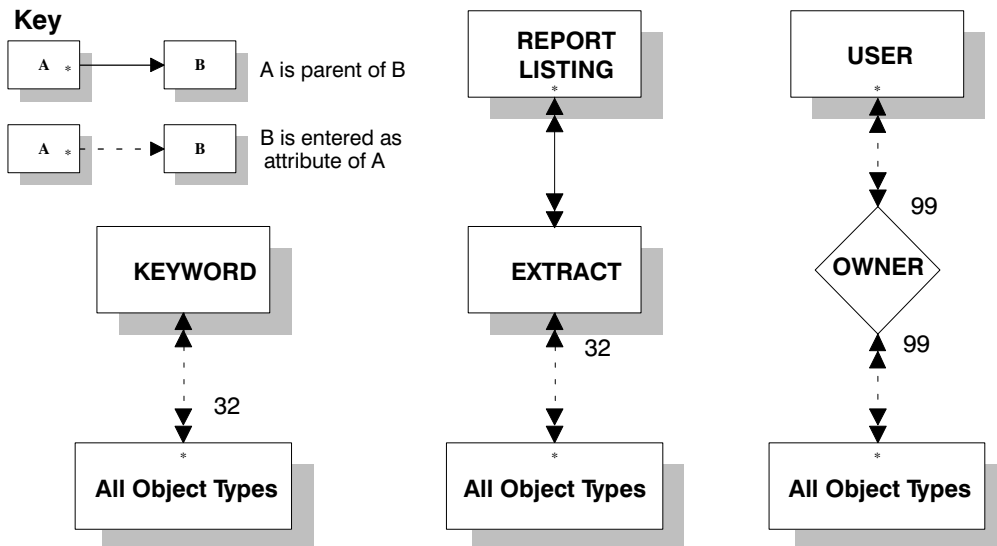
Objects of type *verification* document the processing rules for validating field values.

Chapter 23: Virtual Machine page 444

Together with objects of type *network*, objects of type *virtual machine* document the hardware and operating system environment of a data processing system.

GENERAL INFORMATION

The metastructure of the Predict data dictionary is illustrated below. Additional object types and association types can be defined with Metadata Administration functions. These objects are referred to as *User Defined Object/Association Types* or *User Defined Entities* (UDEs). See Chapter **Metadata Administration** in the *Predict Administration Manual*.



Global Attributes

The following attributes apply to all predefined and user-defined object types.

Object ID

Each object in Predict is identified by its ID. This ID must be unique for objects of the same type.

Note:

Field objects can have the same ID if they belong to different files.

To change the ID of an object, use the function *Rename*. See **Renaming Objects** in Chapter **Maintenance** in the *Predict Reference Manual*.

Naming Conventions

IDs of all objects apart from *verification* are checked against the following naming conventions. (IDs of verifications are checked against Natural naming conventions.)

- The ID of all object types except *user* can be up to 32 characters long.
- Objects of type *user* can have IDs of up to 8 characters.
- The maximum length of object IDs (both for predefined and user-defined object types) can be specified with the metadata administration function *Modify object type*. See Chapter **Metadata Administration** in the *Predict Administration Manual* for more information.
- There is no minimum length for object IDs: one and two character IDs are also possible.
- An object ID must start with a letter (A – Z or a – z).
- The subsequent characters must be alphanumeric, i.e.
 - letters A – Z or a – z
 - digits 0 – 9
 - any special character except *blank*, *asterisk*, *comma*, *question mark*.
 - Up to 20 additional disallowed characters can be specified with the Metadata Administration function *Modify object type*. See Chapter **Metadata Administration** in the *Predict Administration Manual* for more information.
 - The Predict administrator can specify with the parameter *General Defaults>Miscellaneous>Upper/lower case* whether alpha characters in object IDs are converted to upper case. Use of lower-case letters for object IDs is not recommended.

Naming Conventions for Standard Files (File Type Z)

Predict functions which process standard files (file type Z) are considerably faster if the first five characters of each standard file ID are unique.

Naming Conventions for Natural

If the object ID is also to be referenced by a Natural subsystem, the Natural naming conventions should also be observed.

SQL Naming Conventions

Naming conventions for SQL objects are given in Chapter **File** in the Manual *Predefined Object Types in Predict*.

Naming Conventions for Extracts

The following extracts are added automatically with the Coordinator:

- #SAG-TRANSFER
- #SAG-ERROR

See the *Predict Coordinator Manual* for more information. These IDs are reserved.

Copy ID

With most object types, this parameter is used with the *Copy* function for the ID of the new object to be created.

For object type *field* and *file*, this parameter is also used by other functions. See page 94 and page 180 respectively.

In Object

With many object types, a parent object can be specified. For some object types, a parent object is mandatory.

Restrictions

Restrictions are available in every maintenance, retrieval or active retrieval menu. You can limit the selection of objects for processing using a combination of the following:

- **Keywords**
Up to five keywords can be specified. See **Relating Objects Logically** in Chapter **Overview of Predict** in the Manual *Introduction to Predict* and Chapter **Keyword** in the Manual *Predefined Object Types in Predict*.
- **Owner**
You can restrict the retrieval operation to objects that are assigned to a particular owner. See **Relating objects logically** in Chapter **Predict Overview** in the Manual *Introduction to Predict*.
- **Extract**
You can restrict the retrieval operation to objects that are contained in a specified extract. See Chapter **Extract** in the Manual *Predefined Object Types in Predict*.
- **String**
You can restrict the retrieval operation to objects whose abstract, extended description, rules or ID contains the specified string.
- **Date**
Retrieval operations can also be restricted by the parameter *AND from date*: only objects that were added or modified after a given date are evaluated.

See Chapter **Retrieval** in the *Predict Reference Manual* for more information.

Keys

Up to 32 keywords can be assigned to any Predict object, including keywords.

- The keywords, separated by the current input delimiter character, can be specified in the main *Add / Modify* screen. The input delimiter character is defined by the Natural *GLOBALS* command *ID* parameter.
- A keyword must exist as a Predict object before it can be assigned to another object. If you specify a keyword that is not defined in Predict, a *Modify Keyword* window appears in which you can enter a valid keyword. Use asterisk notation to display a range of keywords for selection. Mark the keyword(s) you wish to select with any non-blank character or use cursor selection.
- An asterisk before the *Zoom* field indicates that more keywords have been specified than can be displayed on one line. In this case, enter *Y* here to modify existing keywords or add new keywords.

See Chapter **Keyword** in the Manual *Predefined Object Types in Predict* for more information.

Abstract

Each object in Predict can have an abstract providing short comments on the object.

- An abstract can have up to 16 lines of up to 30 characters.
- Abstracts can contain upper and lower-case letters. If the general default parameter *Miscellaneous > Upper/Lower case > Abstract* is set to *U*, all alphabetic characters are converted to upper-case.
- An abstract can be added, removed or modified whenever the *Add*, *Copy* or *Modify* function is used to maintain an object. The number of abstract lines displayed in the *Add/Copy/Modify* screen depends on the object type. Enter *Y* in the *Zoom:* field to display the maximum 16 lines.

Abstract Editor Commands

The following line commands are available for processing abstracts:

- .c Copy one line.
- .d Delete one line.
- .i Insert three lines.
- .j Join line with next line.
- .s Split line at cursor position.

These line commands are introduced by the escape character defined in the Natural parameter module NATPARM.

EDIT Line Options

Most object types in Predict have the following options in the EDIT line at the bottom of every *Add/Copy/Modify* screen. An asterisk before any option indicates that attributes of the respective type exists.

EDIT Owner	Y	Edit owner list.
EDIT Descr.	Y	Edit description. The editor called depends on the environment in which you are working and various profile parameters.
EDIT File list	Y	Edit file list. The Predict Link Editor is invoked.

See Chapter **Editors in Predict** in the *Predict Reference Manual*.

Note:

All type-specific options in the EDIT line (for example *EDIT Expr.* for fields) are described in the respective chapter of this manual.

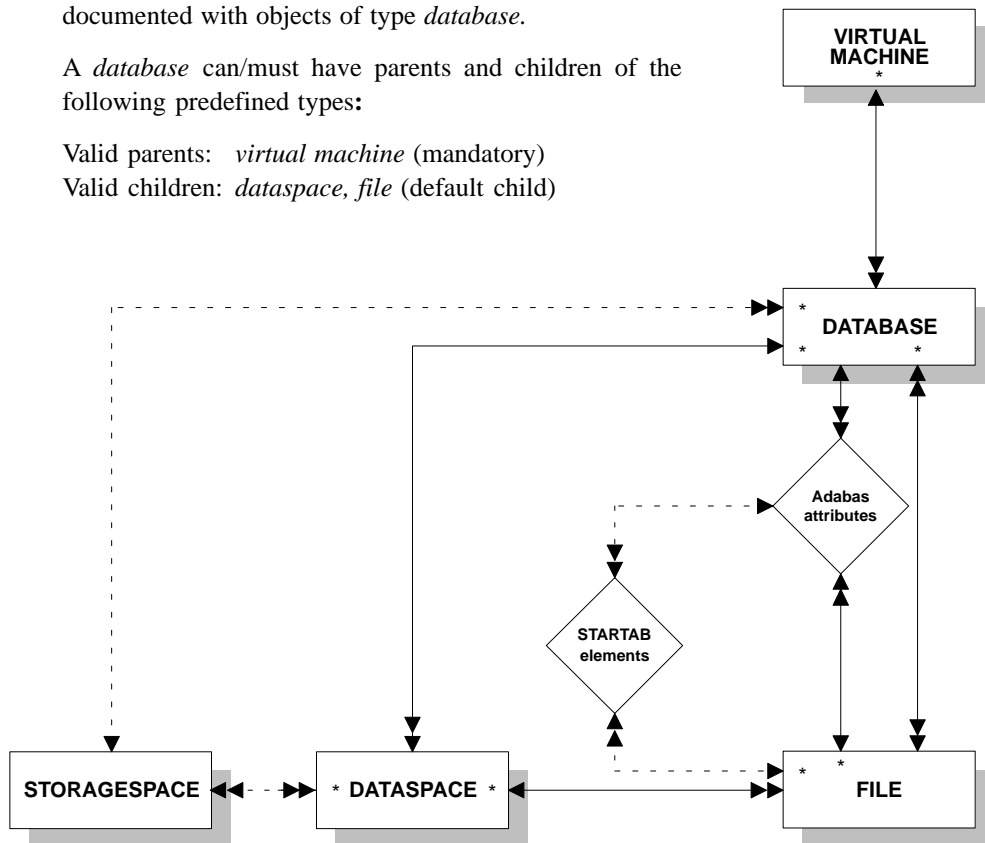
DATABASE

Databases and data storage systems of different types are documented with objects of type *database*.

A *database* can/must have parents and children of the following predefined types:

Valid parents: *virtual machine* (mandatory)

Valid children: *dataspace*, *file* (default child)



How this Chapter is Organized

- **The Database Maintenance Menu** page 14
Defining Basic Attributes of Databases
- **Documenting Databases of Different Types** page 18
 - Adabas C, page 18
 - Adabas SQL handler,
 - Conceptual, General SQL handler, Entire System Server, page 22
 - DB2, page 23
 - RMS handler, rdb Schema, Other handler, page 25
 - IMS, page 26
 - Target node, page 27
 - VSAM handler, page 28
 - Other SQL Databases, page 29
- **Database Specific Maintenance**
 - Purge Database, page 30
 - Rename/Renumber/Retype Database, page 32
 - Special Functions for Editing the File List of a Database, page 37
- **Database Retrieval** page 38
 - Database Retrieval Specific Parameters, page 38
 - Database-specific Retrieval Functions, page 39
 - Explode IMS Databases (Code I), page 39
 - Databases with children (child type *File*, output option *Adabas size=Y*), page 39
 - Layout of Database Lists, page 40
 - Valid Output Options for Database Retrieval, page 41

The Database Maintenance Menu

The *Database Maintenance* menu is called with function code *M* and object code *DA* in a Predict *Main Menu* or the command MAINTAIN DATABASE.

```

10:26:15          ***** P R E D I C T  4.1.1  *****          1999-01-30
Plan  10          - (DA) Database Maintenance -          Profile JCA

Function                                Function
A  Add a database                        D  Display database
C  Copy database                        L  Link children
M  Modify database                      O  Edit owner of a database
N  Rename/renumber/retype database     S  Select database from a list
P  Purge database                      W  Edit description of a database

Function .....
Database ID .....                      Database of type .*
Copy ID .....                          Database number ...
in virtual machine

Restrictions ....*   Profile JCA ,used   Child type .....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Menu  Canc  S-fi  E-el  M-pr  Print Impl  Last  FLIP  PROF  Next

```

Parameters

Note:

Parameters not listed here are described under **Global Attributes**, page 6.

Function	<p>All standard maintenance functions are described in Chapter Maintenance in the <i>Predict Reference Manual</i>.</p> <p>The functions <i>Rename/renumber/retype database</i> and <i>Purge</i> are described below, page 30.</p>																																
Database of type	<p>With the <i>Select</i> function, a database type can be specified as selection criterion.</p> <p>The <i>Add</i> and <i>Copy</i> functions pass the type to the <i>Add/Copy database</i> screen.</p> <p>Note that type <i>I</i> (IMS) is not valid for the <i>Add</i> and <i>Copy</i> functions.</p> <p>Enter an asterisk to display a selection window with the valid database types for a particular function in your environment. The list below shows all valid database types.</p> <table><tr><td>A</td><td>Adabas C</td><td>O</td><td>ORACLE handler</td></tr><tr><td>B</td><td>Adabas D handler</td><td>P</td><td>Entire Syst. Server</td></tr><tr><td>C</td><td>Conceptual</td><td>Q</td><td>Adabas SQL handler</td></tr><tr><td>D</td><td>DB2</td><td>R</td><td>RDB handler</td></tr><tr><td>E</td><td>Gen. SQL handler</td><td>T</td><td>Target node</td></tr><tr><td>H</td><td>Other handler</td><td>V</td><td>VSAM handler</td></tr><tr><td>I</td><td>IMS</td><td>X</td><td>INFORMIX handler</td></tr><tr><td>J</td><td>INGRES handler</td><td>Y</td><td>SYBASE handler</td></tr></table>	A	Adabas C	O	ORACLE handler	B	Adabas D handler	P	Entire Syst. Server	C	Conceptual	Q	Adabas SQL handler	D	DB2	R	RDB handler	E	Gen. SQL handler	T	Target node	H	Other handler	V	VSAM handler	I	IMS	X	INFORMIX handler	J	INGRES handler	Y	SYBASE handler
A	Adabas C	O	ORACLE handler																														
B	Adabas D handler	P	Entire Syst. Server																														
C	Conceptual	Q	Adabas SQL handler																														
D	DB2	R	RDB handler																														
E	Gen. SQL handler	T	Target node																														
H	Other handler	V	VSAM handler																														
I	IMS	X	INFORMIX handler																														
J	INGRES handler	Y	SYBASE handler																														
Database number	<p>For the <i>Add</i> and <i>Copy</i> functions: the database number can be specified here. This number will be passed to the <i>Add a database</i> or <i>Copy database</i> screen. See description of the parameter <i>Physical database number</i>, page 17.</p> <p>For the <i>Select</i> function: a database number can be specified as an additional selection criterion.</p>																																
Child type	<p>For function <i>Link children</i>: objects of this type are to be linked to the database.</p> <p>Valid values: <i>dataspace</i>, <i>file</i> and <i>user-defined</i>.</p>																																

Defining Basic Attributes of Databases

The *Add a database* screen is used by the functions *Add* and *Copy*.

Depending on the database type, one or several type-specific screens follow.

Subsequent screens and their input fields are described in sections below.

Parameters applying to all types of databases are described below.

[illegible]

Attributes

Database type	See page 15 for a list of possible types.
in virtual machine	Predict virtual machine object documenting the hardware and operating system environment of the database. See also Defining the Distribution of Data in Predict in Chapter Adabas Star in the Manual <i>Predict and Other Systems</i> .

ADASTAR parameter

Use of the database with respect to the distribution of data with Adabas Star.

I

Isolated

Adabas Star is not used. The database is isolated.

L

Local

The database cannot be accessed using Entire Net-work.

Y

Translator

Adabas Star is used. The database contains the STARTAB. Only valid for Adabas C databases.

N

No Translator

Adabas Star is used but the database does not contain the STARTAB table. Only valid for Adabas C databases.
See **Defining the Distribution of Data** on page 40 for a detailed description of the meaning of the *ADASTAR parameter*.

Physical database number

Valid values depend on database type:

Database Type	Range of Database Numbers
B, E, J, O, Q, R, X, Y	1 – 255
A, H, M, P, T, V	1 – 65535
Others	not applicable

Documenting Databases of Different Types

Database Type A (Adabas C)

```
10:31:03          ***** P R E D I C T  4.1.1  *****          1999-01-30
                        - Add a Database -

Database ID ..... JCA-DA-NEW
Type ..... ADABAS, Isolated
Physical DBnr ... 244
in virtual machine .. HOME
Keys ..                                           Zoom N

ADABAS attributes                                NATURAL file numbers
  Reflective database .. N (Y,N)                System file (FNAT) ...
  Maximal files .....                          NAT-Security (FSEC) ..
  Checkpoint file .....                        PREDICT (FDIC) .....
  ADABAS security .....
  Size of RABN .....*
  ADASTAR access only .. N

Abstract      Zoom: N

EDIT:   Owner: N   Desc: N   File list: N   MORE:   Size: N
```

Note:
Attributes that are not in the table below are described in the section **Defining Basic Attributes of Databases**. See page 16.

Attributes

Adabas attributes		
Reflective database	Y	A screen for defining the physical attributes of the reflection of a database is displayed (see Specifying the Size of an Adabas C Database , page 20). For further information on reflective databases see the description of the ADAREF Utility in the <i>Adabas Utilities Manual</i> .

Maximal files	<p>Number of files permitted in the database (ADADEF parameter MAXFILES).</p> <p>If <i>isolated database=Y</i>, this number must either be 0 or at least 5 but not more than 5000.</p> <p>If <i>isolated database=N</i>, this number must either be 0 or be at least 5 but not more than 5000.</p>
Checkpoint file	<p>The number of the Adabas C file which contains checkpoint information for the database.</p> <p>Predict automatically creates a data dictionary object with the file ID SAG-ADA-CHECKPOINT for this file.</p>
Adabas security	<p>The number of the Adabas C file which contains Adabas security information for the database. Predict automatically creates a data dictionary object with the file ID SAG-ADA-SECURITY for this file.</p>
Size of RABN	<p>Specifies the length of RABNs in the database.</p> <p>0 not specified</p> <p>3 3 Byte for 24-bit RABNs</p> <p>4 4 Byte for 31-bit RABNs</p>
ADASTAR access only	<p>Y If the attributes of the database are such that files in the database can only accessed using Adabas Star.</p> <p><i>ADASTAR access only</i> is set by Predict.</p> <p>If <i>N</i>, it can be set to <i>Y</i> with the <i>Rename/Retype/ Renumber</i> function (code <i>N</i>). See page 35.</p>
Natural file numbers	
System file (FNAT)	The number of the Natural system file.
NAT Security (FSEC)	The number of the Adabas C file which contains Natural Security information.
Predict (FDIC)	The number of the Adabas C file which contains the dictionary data.
Additional Option in the EDIT Line	
MORE Size	<p>Y An additional screen is displayed for specifying the physical size of the database.</p> <p>See Specifying the Size of an Adabas C Database below.</p>

Specifying the Size of an Adabas C Database

Physical properties of a database (device types and sizes of the datasets containing the Adabas C ASSO, DATA, WORK, SORT and TEMP) can be defined in the screen below.

The screen is displayed by setting the parameter *Size* in the EDIT line of the *Add/Copy/Modify Database* menus to *Y*.

```

10:31:57          ***** P R E D I C T  4.1.1  *****          1999-01-30
                        - Add a Database -
Database ID ..... JCA-DA-NEW                      Added 1999-01-30 at 10:29
Type ..... ADABAS Isolated                          by JCA
Physical DBnr ... 244
----- Database primary sizes -----
              Number of      Alternate RABN
              *Device      Cylinder  RABN      Start      End
ASSO R1
      R2
      R3
      R4
DATA R1
      R2
      R3
      R4
WORK R1
      R2
SORT R1
      R2
TEMP R1

EDIT:  Owner: N   Desc: N   File list: N MORE:  MIRROR: N   ASSO: N   DATA: N

```

Rules for Defining the Size of a Database

- If *Reflective database* is set to *Y*, a similar screen is displayed in which the device type and size of the Adabas C Associator file, the Adabas C Data Storage file and the Adabas C workfile for the database mirror can be specified.
- If the device type and the size in RABNs (relative Adabas block numbers) of each extent is specified, Predict calculates and displays the equivalent size in cylinders, beginning with a *greater than* sign (>) unless the number of cylinders is exactly equivalent.

- If the size is specified only in cylinders, Predict calculates and displays the equivalent size in RABNs. Adabas C does not use the first track of the first extent of the Associator, Data Storage and workfiles. In these extents, the number of RABNs is therefore smaller than the number of blocks contained by the specified number of cylinders. The start and end of the range of alternate RABNs can also be specified.
- Four extents for ASSO and DATA (R1 – R4) can be defined in the above screen. To define more extents (up to 16) the parameter *ASSO* and/or *DATA* in the EDIT line of the screen have to be set to *Y*.

Note:

See the *Adabas Administration Manual* for detailed information on the topic.

Parameters

Device	Devices are identified with a four-letter code that must have been defined with the function <i>Adabas C device types</i> in the <i>Special functions</i> menu. If a device type is changed, the change should also be made in each file objects that is linked to the database.
Cylinder	The number of cylinders of the specified device that are occupied by the specified extent of the specified database.
Number of RABN	The number of RABNs (relative Adabas block numbers) of the specified device that are occupied by the specified extent of the specified database.
Alternate RABN	The first and last RABN that were reserved on the specified device as alternate RABNs for the specified database. Alternate RABNs can be defined by using either the ADADEF utility or – for a reflective database – the ADAREF utility. For further information see the <i>Adabas Utilities Manual</i> .

Options in the EDIT Line

EDIT ASSO	Y	If more than four extents are to be defined.
EDIT DATA	Y	If more than four extents are to be defined.

Database Types C, E, P (Conceptual, General SQL Handler, Entire System Server Nodes)

The following screen is displayed when adding, modifying or copying databases of the types *C*, *E* and *P*.

10:33:21

***** P R E D I C T 4.1.1 *****

1999-01-30

- Add a Database -

Database ID JCA-DA-C

Type Conceptual

Keys ..

Zoom: N

Abstract

Zoom: N

All parameters are described in section **Defining Basic Attributes of Databases**, page 16.

Database Type D (DB2)

The following attributes apply to databases of type *D*. Attributes not listed here are described on page 16.

```
17:23:06          ***** P R E D I C T  4.1.1  *****          1999-07-24
                                - Add a Database -

Database ID ..... JCA-DB2
Type ..... DB2
in virtual machine .. HOME
Keys ..                                           Zoom N

DB2 attributes
DB2 name .....
Default storagespace ....*
Buffer pool .....*
SQL type .....* DB2
DB2 ROSHARE parm .....*   Not shared
Data sharing group member.
CCSID .....*   (none)

Abstract      Zoom: N
```

Attributes

DB2 Attributes	
DB2 name	The name of the database in DB2.
Default storagespace	DB2 tables of the database will be implemented in this storagegroup if no other storagegroup is explicitly specified.
Buffer pool	The buffer pool of the database. Enter an asterisk for valid values.
SQL type	Valid values: DB2 SQL/DS

DB2 ROSHARE parm	<p>Indicates how the database will be shared using shared read-only data. This parameter determines the ROSHARE clause generated for a CREATE TABLE statement generated from this database object.</p> <p><i>blank</i> ROSHARE clause is not generated. Database will not be shared.</p> <p>O Owner. Clause ROSHARE OWNER is generated.</p> <p>R Read. Clause ROSHARE READ is generated.</p> <p><i>Note:</i> If this parameter is set to <i>R</i>, you must specify parameter <i>OBid</i> for tables contained in this database. See page 221.</p> <p>See your DB2 documentation for more information.</p>
Data sharing group member	<p>Name of the member of the data sharing group.</p> <p>Leave blank or specify name with up to eight characters (letters A-Z, digits 0-9 and special characters \$, # and @).</p>
CCSID	<p>Defines the encoding scheme of the database.</p> <p><i>blank</i> not specified.</p> <p>A ASCII.</p> <p>E EBCDIC.</p>

Database Types *Q*, *M*, *R*, *H* (Adabas SQL Handler, RMS Handler, rdb Handler, Other Handler)

Database type *Q* is used to document databases of type *Adabas SQL handler*. See Chapter **Adabas SQL Server** in the Manual *Predict and Other Systems* for more information.

Database type *M* is used to document RMS databases; database type *R* is used to document rdb databases. See also Chapter **RMS** and Chapter **rdb** NO TAG in the Manual *Predict and Other Systems*.

Database type *H* is used to represent database handlers, such as USER-DB, SESAM, DL1, WIZZARD, TRS etc. Database type *other handler* can be used to reserve a database number (prevent it from being used by Adabas C).

```
11:03:04          ***** P R E D I C T  4.1.1  *****          1999-01-30
                        - Modify Database -
Database ID ..... JCA-DA-M                      Added 1997-01-30 at 10:51
Type ..... RMS Handler                          by JCA
Physical DBnr ... 123
in virtual machine .. JCA-VM1
Keys ..                                           Zoom N

Abstract      Zoom: N
```

Parameters

Physical DBnr For database type *RMS Handler*: the database number must be declared in NATPARM as an RMS database number if DDMs for RMS files contained in the database are to be generated. See table on page 17 for range of permitted values.

Database Type I (IMS)

IMS databases cannot be added with the *Add a database* function. To create an IMS Database object in Predict, an existing IMS database must be incorporated with the INCORPORATE NDB function.

```
11:20:27          ***** P R E D I C T  4.1.1  *****          1999-01-30
                                - Modify Database -
Database ID ..... RSH-CUSTOMER          Added 1998-10-05 at 16:11
Type ..... IMS                          by RSH
in virtual machine ..
Keys ..                                  Zoom N

IMS attributes
  IMS or DL1 ..... IMS
  IMS name .....
  IMS type ..... PHYSICAL

Abstract      Zoom: N
This database was incorporated
from NDB: CUSTOMER
on 1998-10-05

EDIT:   Owner: N   Desc: N * File list: N
```

The following attributes apply to databases of type *I*. For for attributes that are not in the table, see section **Defining Basic Attributes of Databases**, page 16.

Attributes

IMS attributes	
IMS or DL1	The kind of database. Valid values: IMS DL1
IMS name	The name of the database in IMS.
IMS type	The type of the database in IMS. Valid values: LOGICAL PHYSICAL.

Database Type *T* (Target Node)

Database type *T* is used to represent database nodes entered in the ID table of an SVC which cannot be documented with a corresponding database type: BROKER, NATURAL GLOBAL BUFFER POOL etc.

This type of database is used to reserve the corresponding database number and thus prevent this number being used for an Adabas C database.

Databases of type *T* are defined in two screens:

```
11:29:32          ***** P R E D I C T  4.1.1  *****          1999-01-30
                        - Add a database -

Database ID ..... JCA-DA-T

Database type .....* T Target Node
in virtual machine .....* HOME
ADASTAR Parameter.....* I Isolated
Physical database number ..* 135
```

```
11:23:47          ***** P R E D I C T  4.1.1  *****          1999-01-30
                        - Add a Database -

Database ID ..... JCA-DA-T
Type ..... Target Node, Isolated
Physical DBnr ... 135
in virtual machine .. HOME
Keys ..
Zoom N

Abstract      Zoom: N
```

Attributes

Attributes not listed here are described on page page 16.

- | | |
|--------------------------|--|
| ADASTAR parameter | Must be specified for databases of this type. Valid values:
I Isolated
L Local |
| Physical database number | The physical database number must be in range 1–65535. |

Database Type V (VSAM Handler)

Database objects of type *V* are used to collect all definitions of VSAM clusters which are accessed by the same Natural VSAM handler. The database number defined in a database object of type *V* is used by the GENERATE DDM function.

Databases of type *T* are defined in two screens:

```
11:57:30          ***** P R E D I C T  4.1.1  *****          1999-01-30
                        - Add a database -
Database ID ..... JCA-DA-V
```

```
Database type .....* V VSAM Handler
in virtual machine .....* HOME
ADASTAR Parameter.....* L Local
Physical database number ..*
```

```
11:59:18          ***** P R E D I C T  4.1.1  *****          1999-01-30
                        - Add a Database -
```

```
Database ID ..... JCA-DA-V
Type ..... VSAM Handler
Physical DBnr ... 199
in virtual machine .. HOME
Keys ..
```

Zoom N

```
Abstract      Zoom: N
```

Attributes

Attributes not listed here are described on page page 16.

ADASTAR parameter Must be *local* for databases of this type.

Physical database number The physical database number must be in the range from 1 – 65535.

Other SQL Database Types

The screens used to maintain database objects of the following types are the same as for VSAM databases. See page 28. The physical database number must be less than or equal to 254.

J INGRES Handler

O ORACLE Handler

X INFORMIX Handler

Y SYBASE Handler

B Adabas D Handler

Database-Specific Maintenance

Maintenance functions applying to databases are called from the *Maintain Database* menu that is called with the command MAINTAIN DATABASE or with function code *M* and object type code *DA* in a Predict *Main Menu*.

This section covers the following topics:

- *Purge Database*
- *Rename/Renumber/Retype Database*, page 32
- Special functions for editing the file list of a database, page 37

Purge Database (Code *P*)

Predict objects of type *Database* are deleted with the *Purge* function (code *P*). You have two purge options, *Delete* and *Scratch*.

DELETE

The DELETE option applies to all database types apart from *IMS*. The following objects are deleted:

- the database object
- all links to parent and child objects.

Rules which apply to the individual database types are given below.

SCRATCH

The SCRATCH option deletes the following objects:

- files in this database and the related userviews
- fields of these files
- generated code of these files
- file relations based on these files
- links to/from the scratched objects.

Two lists will be displayed before a database is purged:

- A list of objects and generated code that will not be deleted because they are used in some other object which will not be deleted. This list will only be displayed if the *Purge mode* option in the session profile is set to *Y*. See **Customizing Predict with Profiles** in Chapter **The User Interface** in the Manual *Introduction to Predict*.
- A list of objects generated code that will be deleted.

Confirmation of the purge operation is then requested. A list of all deleted objects and links will be displayed after execution of the delete operation.

Database-specific Rules

For Database Objects of Type *Adabas C*

- A *Purge* operation is not executed if the database and files in the database are implemented.
- The *Delete* operation purges a database object and all links to related objects. All Adabas C attributes for files which are linked to this database are purged or changed to default if the file is not linked to another database.
- File objects for which DDMs or table/cluster descriptions exist will not be purged.

For Database Objects of Type *SQL*

- A *Purge* operation is not executed if the database and files in the database are implemented.
- File objects for which DDMs or table/cluster descriptions exist will not be purged.

For Database Objects of Type *IMS*

- The *Purge* function will not be executed if UDFs exist for the IMS files.
- *Delete* is not available because Predict regards an IMS database object and the files contained in it as an integral unit.

Rename/Renumber/Retype Database (Code N)

```

12:48:50          ***** P R E D I C T  4.1.1  *****          1999-01-30
                                - Rename Database -
Database ID ..... JCA-DA-A                      Added 1998-01-30 at 12:48
Database type ... ADABAS C                        by JCA

Enter new values

Database ID ..... JCA-DA-A
Database type .....* A ADABAS C
in virtual machine ..* HOME
ADASTAR parameter ...* N No Translator
Physical DBnr .....* 1234
ADASTAR access only .. Y (Y,N)

Enter '.' to return to menu.

```

This function can be used to change

- Database ID
- database Type
- the virtual machine of the database
- the *ADASTAR* parameter
- physical database number
- the *ADASTAR access only* flag.

Depending on the database to be processed, messages indicating the possible Rename/Renumber/Retype options are displayed at the bottom of the screen.

General Rules

- Changes to database attributes are also applied to file objects if applicable. For example: if a database is linked to another virtual machine, existing STARTAB elements of files linked to the database are adapted accordingly.
- Special rules apply when renaming/renumbering/retyping databases that are connected to implemented databases. Connecting documentation and external objects is described in Chapter **Handling of External and Documentation Objects** in the Manual *External Objects in Predict*.

Changing the Database ID

The new database ID must not already exist in the dictionary.

Changing the Database Type

- If files linked to the database are connected to implemented files, the database type and the database number (*DBnr*) cannot be changed.
- For a database of type *C* (conceptual) the following rules apply:
 - If all files linked to the database have the same type, the database type can be changed to this type.
 - If files linked to the database have different types, the database type must not be changed.
- A Database of type *A* (Adabas C) cannot be changed to type *C* (conceptual) if the database contains a partitioned master file or partitioned replicated file with a counterpart linked to another database.
- All other database types can be changed to type *C* (conceptual) without restrictions.

Linking the Database to another Virtual Machine

The following rules apply to Adabas C databases:

- Old virtual machine and new virtual machine are in the **same network**:
the change is applied to the database and all files linked to the database.
- Old virtual machine and new virtual machine are in **different networks**:
new STARTAB elements are created and/or existing STARTAB elements are purged.
Additional confirmation is requested when purging STARTAB elements (as shown in the screen below).

```

15:17:38          ***** P R E D I C T  4.1.1  *****          1999-01-22
                        - Rename Database -
Database ID ..... HEB-NO-TRANS          Modified 1998-09-29 at 15:17
Database type ... ADABAS                  by JPE

      +-----+
Enter n !          A T T E N T I O N          !
      !          !          !          !
Datab ! Old and new virtual machine are not in the !
Datab ! the same network.          !
in vi ! STARTAB elements will be purged.          !
ADAST ! old NW: HEB-NW-TEST          !
Physi ! new NW: HEB-NW          !
ADAST !          !          !
      ! Do you want to continue N (Y/N)          !
Enter ' +-----+

File with phys. ADASTAR type partitioned or replicated found.
File with STARTAB element found.

```

- If the Database is connected to an implemented database or it contains a partitioned master file or partitioned replicated files with a counterpart linked to another database, the new virtual machine must be in the same network as the old virtual machine.
See also the description of the message *Partitioned master and replicate in different database* on page 36.

Non-Adabas databases can be linked to another virtual machine without restrictions.

Changing the ADASTAR Parameter

The following rules apply:

- Changing from ADASTAR parameter *I* (isolated) or *L* (local) to *Y* (translator database) or *N* (no translator) is not possible if a replicated or master file for Entire Transaction Propagator is linked to the database.
- Changing from ADASTAR parameter *Y* (translator database) or *N* (no translator) to *I* (isolated) or *L* (local) is not possible if:
 - files with STARTAB elements are linked to the database, or
 - files with ADASTAR type are linked to the database.

Changing the Database Number

The following rule applies:

- If files linked to the database are connected to implemented files, the database number (*DBnr*) cannot be changed.

Changing the Parameter *ADASTAR Access Only*

The *ADASTAR Access Only* flag indicates whether Adabas Star is required to access files in a database. When creating a database object, Predict set this flag to *Y* or *N* (according to the attributes of the database). The following rule applies:

- Setting the flag from *N* to *Y*:
Only possible for databases of type *Y* (translator database) or *N* (no translator). *ADASTAR* attributes for files will be created if they do not already exist.

Messages

If prerequisites for renaming/renumbering/retyping databases are not met, one of the following messages is issued.

Implemented file exists in the database

If files linked to the database are connected to implemented files, the database type and the database number (*DBnr*) cannot be changed.

Partitioned master and replicate in different database

Partitioned replicated files and partitioned master files are connected by *DBnr* and *Fnr*. If the partitioned replicated file and the corresponding partitioned master file are not linked to the same database, the following restrictions apply:

- The database must not be linked to a virtual machine that belongs to a different network.
- The database type must not be changed if the partitioned master file is linked to the database that has been renamed.

File with physical ADASTAR type 'partitioned' or 'replicated' found

File with STARTAB element found

At least one file defined for use with Adabas Star (ADASTAR type or STARTAB element is specified) is linked to the database. The ADASTAR parameter can therefore only be changed to *Y* (translator database) or *N* (no translator); the database type can only be changed to *C* (conceptual).

File with PROPAGATOR type 'master' or 'replicated' found

At least one replicated or master file for ENTIRE TRANSACTION PROPAGATOR has been found. The ADASTAR parameter can therefore not be changed to *Y* (translator database) or *N* (no translator); the database type can only be changed to *C* (conceptual).

Different File types in the database

Files of different types are linked to the database. Database therefore must be of type *C* (conceptual) and cannot be changed to another type.

Special Functions for Editing the File List of a Database

The following line commands apply when editing the file list of a database.

Editor Commands

SORT LOG	Sort the list of file IDs into ascending order of their logical file numbers.
SORT PHY	Sort the list of file IDs into ascending order of their physical file numbers.

Line Commands

.A	Calls the <i>Modify Adabas C Attributes</i> screen for the file.
.E	Calls the <i>Add file</i> screen for files that have just been added to the file list of the database (dummies) or the <i>Modify file</i> screen for Files that already exist.
.T	Calls the <i>Modify STARTAB Element</i> screen for the file.

Database Retrieval

Retrieval functions applying to database objects are called from the *Database Retrieval* menu that is called with the command RETRIEVE DATABASE or with function code *R* and object type code *DA* in a Predict *Main Menu*.

This section covers the following topics:

- Database-specific retrieval parameters, page 38
- Database-specific retrieval functions
 - Explode IMS databases (Code *I*), page 39
 - Databases with children (child type *file*, output option *Adabas size=Y*), page 39
- Layout of database lists, page 40
- Output options, page 41

Note:

Standard retrieval functions are described in Chapter **Retrieval** in the *Predict Reference Manual*.

Database-Specific Retrieval Parameters

Parameters

Database of type	Limits the scope of the function to databases of a certain type. Enter an asterisk to display possible values at your site or see complete list of database types on page 15.
Database number	Limits the scope of the function to databases with the number specified.

Database-Specific Retrieval Functions

Explode IMS Database (Code I)

Shows the hierarchical structure of an IMS/DL1 database. The level number before the file ID shows the level of the IMS/DL1 segment in the hierarchy. This function is only applicable to databases of type *I*. Command: EXPLODE DATABASE.

Databases with children
(with Child Type *File* and Output Option *Adabas size=Y*)

The following output is produced with function *Display databases with children* if you specify child type *file* and output option *Adabas size=Y* for databases and files of type *Adabas C*.

```
13:16:02          ***** P R E D I C T 4.1.1 *****          1999-02-14
                    - Display Database with Children -

Database ID ..... DA-WITH-FILE
Type ..... ADABAS C,No Transl      Added 1998-02-14 at 10:25 by ARH
Physical DBnr ..... 57              Modified 1998-02-14 at 10:42 by ARH
-----
ADABAS attributes          NATURAL file numbers
Reflective database .. N    System file (FNAT) ...
Maximal files .....       NAT-Security (FSEC) ..
Checkpoint file .....     PREDICT (FDIC) .....
ADABAS security .....
Size of RABN ..... 0
ADASTAR access only .. N

----- Database primary sizes -----
      Device  Cylinder  Number of RABN  Alternate RABN
ASSO R1  3390      15        4032        Start  End
ASSO R2  3390      14        3780        1200  3200
DATA R1  3380      12        1611        2000  3760
DATA R2  3380      13        1755

Cnt  File ID          Type  Fnr  PDM Impl Other
  1  FI-A-001          A    123
  2  FI-A-002          A    124

----- Summary of sizes -----
Type  Device  No. of RABN  Min. RABN  Undoc. RABN
ASSO  3390    7812    1942    5870
DATA  3380    3366    1205    2161

*** End of report ***
```

The total ASSO and DATA sizes defined for the database are calculated and displayed in column *Summary of sizes/No. of RABN*.

The sum of the sizes of ASSO and DATA for the individual related files is calculated and displayed in column *Min. RABN*.

The number of available RABNs is displayed in the column *Undoc. RABN*, or a message is given indicating that the sizes of the files exceed the size available in the parent database.

The system also checks whether the devices specified for the files are also specified for the parent database.

Layout of Database Lists

The following list format applies when retrieving information on databases with output mode *List*.

13:37:03	***** P R E D I C T 4.1.1 *****	1999-02-14
	- List Database -	

Cnt	Database ID	Type P-DBnr ADASTAR Parm.
15	ARH-X	INFORMIX Handler 24 Local
16	ARH-Y1	SYBASE Handler 22 Local
17	* ARTICLE	IMS 22 Local
18	AZ-PREDICT	ADABAS C 66 Isolated
19	AZ-SAGPRD	DB2 0 Isolated
*** End of report ***		

Meaning of Columns

Database ID	<p>ID of the database object.</p> <p>If the output option <i>Mark implementation</i> is set to <i>Y</i>, implemented objects are marked with an asterisk. For databases, “implemented” means that it is one of the following:</p> <ul style="list-style-type: none"> - of type <i>A</i> and connected to a physical Adabas C database, - of type <i>D</i> and connected to a physical DB2 database - of type <i>P</i> and its database number is defined in the NTDB macro as a Entire System Server database, - or of type <i>I</i>.
Type	The database type. See page 15.
P-DBnr	The physical number of the database.
ADASTAR Parm	<p>Accessibility of Adabas C databases using Adabas Star.</p> <p>Possible values are listed on page 17.</p>

Output Options for Database Retrieval

Retrieval Type	D				B				O				T							
													<i>dummies=Y N</i>				<i>dummies=D P</i>			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																				
Adabas sizes	✓				✓				✓				✓				✓			
Association attributes					✓	✓	✓	✓					✓	✓	✓	✓				
Attributes	✓				✓				✓				✓				✓			
Check expression																				
Composed fields																				
Connecting character						✓								✓						
Cover page	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Description	✓				✓	✓			✓				✓	✓			✓			
Display length																				
Display modifier	✓				✓				✓				✓				✓			
DV-field expression																				
Dummy/Placeholder														✓		✓		✓		✓
Entry points																				
Extract	✓				✓	✓			✓				✓	✓			✓	✓		
Generation layout																				
Adabas C version																				
Language																				
Alignment/sync.																				
Position/Offset																				
Counter length																				
Compiler																				

Retrieval Type	D				B				O				T							
													<i>dummies=Y N</i>				<i>dummies=D P</i>			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓				✓	✓			✓				✓	✓			✓			
Linked verification																				
Mark implementation	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
No. abstract lines	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
Natural options																				
Owner	✓				✓	✓			✓				✓	✓			✓			
With users	✓				✓	✓			✓				✓	✓			✓			
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Procedure code																				
Rules																				
Show implementation	✓				✓				✓				✓				✓			
Sorted by field																				
Subquery																				
Synonyms																				
STARTAB elements																				
Trigger																				
Use Con-form	✓				✓	✓			✓				✓	✓			✓			
User exit	✓				✓				✓				✓				✓			
3GL specification																				

Output Options for Database Retrieval (Continued)

Retrieval Type	U				E				C				I			
Output Mode	D		L		T		X		L		D		T			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																
Adabas sizes	✓															
Association attributes					✓	✓										
Attributes	✓					✓		✓								
Check expression																
Composed fields																
Connecting character						✓		✓				✓				
Cover page	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Description	✓							✓				✓				
Display length																
Display modifier	✓													✓		
Dummy/Placeholder						✓		✓	✓		✓					
DV-field expression																
Entry points																
Extract	✓					✓		✓			✓	✓				
Generation layout																
Adabas version																
Language																
Alignment/sync.																
Position/Offset																
Counter length																
Compiler																
Replace with syn.																

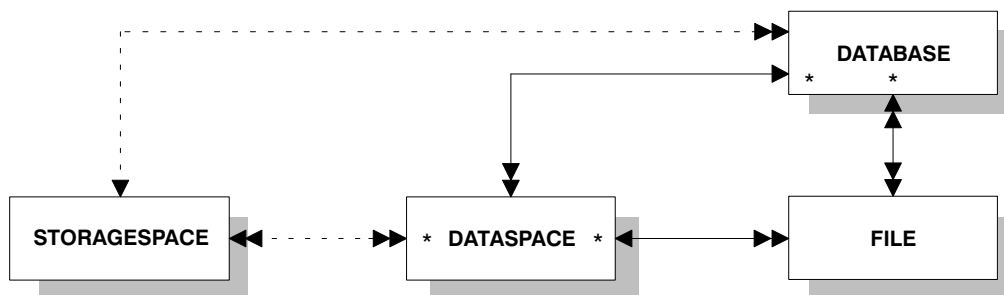
Retrieval Type	U				E				C				I			
Output Mode	D		L		T		X		L		D		T			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓					✓		✓				✓				
Linked verification																
Mark implementation	✓		✓		✓	✓	✓	✓		✓		✓	✓	✓		
No. abstract lines	✓		✓			✓		✓		✓		✓		✓		
Natural options																
Owner	✓					✓		✓				✓				
With users	✓											✓				
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓	✓		✓	✓	✓		
Procedure code																
Rules																
Show implementation	✓															
Sorted by field																
Subquery																
Synonyms																
STARTAB elements																
Trigger																
Use Con-form	✓							✓				✓				
User exit	✓															
3GL specification																

DATASPACE

DB2 table spaces or SQL/DS DBspaces are documented with objects of type *dataspace*.

Note:

DB2 storagegroups are documented with objects of type *storagespace*.



A dataspace can have parents and children of the following predefined types:

Valid parent: database (default parent)

Valid children: file (default child)

How this Chapter is Organized

- **Dataspace Maintenance**
 - The Dataspace Maintenance Menu, page 47
 - The Add/Copy/Modify Dataspace Screen (DB2), page 49
 - The Add/Copy/Modify Dataspace Screen (SQL/DS), page 55
 - Function *Purge Dataspace*, page 56
- **Dataspace Retrieval**
 - Layout of Dataspace Lists, page 57
 - Output Options, page 58

The Dataspace Maintenance Menu

The *Dataspace Maintenance* menu is called with function code *M* and object code *DC* in a *Predict Main Menu* or the command MAINTAIN DATASPACE.

15:34:10

***** P R E D I C T 4.1.1 *****

1999-02-28

Plan 0

- (DC) Dataspace Maintenance -

Profile JCA

Function

Function

A Add a Dataspace

D Display Dataspace

C Copy Dataspace

L Link children

M Modify Dataspace

O Edit owners of a Dataspace

N Rename Dataspace

S Select Dataspace from a list

P Purge Dataspace

W Edit description of a Dataspace

Function

Dataspace ID ARH-DC-3

Copy ID

in database

Restrictions* Profile JCA ,used

Child type*

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---

Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next

Parameters

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Function	Executes one of the maintenance functions. Standard maintenance functions are described in Chapter Maintenance in the <i>Predict Reference Manual</i> . The function <i>Purge</i> is described on page 56.
----------	---

The Add Dataspace Screen

The following screen is displayed for the *Add a Dataspace* function.

16:00:49

***** P R E D I C T 4.1.1 *****

1999-02-28

- Add a Dataspace -

Dataspace ID JCA-DC-2

Dataspace type* S SQL-DS

in database*

Parameters

Dataspace type	D DB2	
	S SQL/DS	
	A second input screen is displayed depending on the type. The screens are described below.	
in database	The ID of the database which contains the dataspace.	
	Applicable to DB2 dataspace.	

The Add/Copy/Modify Dataspace Screen (DB2)

The following screen applies to DB2 dataspace (type *D*).

13:47:48

***** P R E D I C T 4.1.1 *****

1999-09-13

- Add a Dataspace -

Dataspace ID HEB-DC1

Type DB2

in database HEB-DB2

Keys ..

Zoom: N

Dataspace attributes

Tablespace name ..

Nr of partitions .

Buffer pool*

Locksize*

Close option (Y/N)

Lockmax (Y/N)

Lockpart (Y/N)

Maxrows (Y/N)

CCSID*

Member cluster ... (Y/N)

Large (Y/N)

Pages per segment .

Password required . (Y/N)

Abstract

Zoom: N

EDIT: Owner: N Desc: N Files: N

MORE Using/free. N

Parameters

Tablespace name	Name of the table space in DB2.
Nr of partitions	Number of partitions used by the table space (corresponding to the Numparts parameter, max. 254). If 0 is specified, the table space is not partitioned. <i>Nr of partitions</i> must be zero if parameter <i>Pages per segment</i> >0.
Large	Partitions can be defined explicitly or with default values (see parameter <i>Using/free</i> below). Partition definitions are used when generating table spaces from Predict dataspace objects. Identifies a table space as large. Y Yes N No

Buffer pool	Name of the buffer pool to be associated with the table space. Enter asterisk for list of valid values.
Locksize	Locking level for the table space. Valid values: A any level locking P page level locking R row level locking S table space level locking. T table level locking (only valid for segmented <i>DS</i>)
Close option	Y The datasets which support the table space are closed when nobody is using the table space.
Lockmax	The maximum number of pages or row locks an application can hold simultaneously in the table space. Valid values: - <i>blank</i> - SYSTEM - value between 0 and 2,147,483,647. If parameter <i>Locksize</i> is set to <i>S</i> or <i>T</i> , <i>Lockmax</i> must be set to 0.
Lockpart	Partition locking. Valid values: <i>blank</i> not specified Y Yes N No
Maxrows	The maximum number of rows.
CCSID	Encoding scheme. Valid values: <i>blank</i> not specified A ASCII E EBCDIC
Member cluster	The maximum number of rows Valid values: <i>blank</i> not specified Y Yes N No

Pages per segment		How many pages are to be assigned to each segment (parameter SEGSIZE). Zero for table spaces that are not segmented. <i>Pages per segment</i> must be zero if parameter <i>Nr of partitions</i> >0.
Password required	Y	A password must be entered when generating DB2 table spaces.
MORE Using/free	Y	The partitions of the table space are to be defined. The following two options are available: <ul style="list-style-type: none"> - A default definition can be specified (the <i>Using/free clause</i>). The default values are used for partitions that are not defined explicitly. - Individual partitions can be defined. The screens to define individual partitions follow the screen for the definition of the default values.

Default Definition for Partitions

The values specified in the *Definition of using/free clause* section are used as default values for the partition definition.

Partitions can be defined explicitly in subsequent screens. See below.

```

18:03:34          ***** P R E D I C T  4.1.1  *****          1999-07-24
                        - Modify Dataspace -
Dataspace ID .... JCA-DC                      Added 1999-07-24 at 17:50
                                              by JCA

Definition of using/free clause
  VSAM catalog name .....
or Storagespace .....*

Primary attributes
  Free pages .....
  Percentage free .....
  Compress option ..... (Y/N)
  GBPCACHE .....*

Additional for storagespace
  Primary allocation ....
  Secondary allocation ..
  Erase option ..... (Y/N)

EDIT:   Owner: N   Desc: N   Files: N           MORE   Partition: N

```

Parameters

VSAM catalog name	Name of the VSAM catalog containing an entry for the datasets of the table space. Must not be specified if the parameter <i>Storagespace</i> is specified.
Storagespace	Name of the storagespace for the table space documented with the Predict Dataspace object. Must not be specified if the parameter <i>VSAM catalog</i> is specified.
Primary attributes	
Free pages	How often pages are to be left free when loading or reorganizing table spaces or partitions. Max. value is 255. Default is 0, leaving no free pages.
Percentage free	Percentage of each page to be left free.

GBPCACHE	<p>Only relevant in a data sharing environment. Specifies what pages of the table space or partition are written to the group buffer pool. Leave this field blank or enter:</p> <p>C Changed. Only pages that have been changed are written to the group buffer pool.</p> <p>A All pages are written.</p>
Additional for storagespace	
Primary allocation	Primary space allocation for DB2 defined data sets.
Secondary allocation	Secondary space allocation for DB2 defined data sets.
Erase option	<p>Determines if DB2 defined datasets are to be erased when the table space is dropped:</p> <p>N Do not erase datasets (default).</p> <p>Y Erase data sets.</p>

Defining Partitions

Each individual partition can be defined in the *Definition of partitioned Dataspace* section. Two partitions can be defined in one screen. The maximum number of partitions is 254. To modify a specific partition, skip previous definitions by pressing ENTER.

18:25:27

***** P R E D I C T 4.1.1 *****

1999-07-24

- Modify Dataspace -

Dataspace ID ... SMR-NEUER-DATENPACK

Modified 1999-06-13 at 14:59 by SMR

----- Definition of partitioned dataspace -----

Partition 1

VSAM catalog name

or Storagespace*

Primary attributes

Free pages

Percentage free

Compress option (Y/N)

GBPCACHE*

Additional for storagespace

Primary allocation

Secondary allocation ..

Erase option (Y/N)

Partition 2

VSAM catalog name

or Storagespace*

Primary attributes

Free pages

Percentage free

Compress option (Y/N)

GBPCACHE*

Additional for storagespace

Primary allocation

Secondary allocation ..

Erase option (Y/N)

EDIT: Owner: N Desc: N * Files: N

MORE

Partition: Y

Parameters

Partition *n* Identifier of the partition to be defined.

See previous page for a description of all other parameters.

The Add/Copy/Modify Dataspace Screen (SQL/DS)

The following screen applies to SQL/DS Dataspaces (type S).

16:37:47

***** P R E D I C T 4.1.1 *****

1999-02-28

- Add a Dataspace -

Dataspace ID JCA-SQ-2

Type SQL-DS

Keys ..

Zoom: N

Dataspace attributes

Tablespace name

Private dataspace (Y/N)

Size for header

Size for dataspace

Percentage for indices ..

Percentage free

Lock size*

Storage pool number

Abstract

Zoom: N

Parameters

Tablespace name	Identifier of the table space and name of the DBspace in SQL/DS.
Private Dataspace	Y SQL/DS DBspace is private. N Dataspace is public.
Size for header	Number of 4096-byte logical pages reserved for header.
Size for Dataspace	Size reserved for the dataspace.
Percentage for indices	Percentage of the reserved space that can be used for indexes.
Percentage free	Percentage of reserved space to be kept free.
Locksize	Locking level for the dataspace. Valid values: P page S dbspace R row
Storage pool number	Storage pool number. This parameter tells SQL/DS to acquire the dbspace from a specified storage pool.

Dataspace-Specific Maintenance

When maintaining dataspace, only standard maintenance functions are needed. However, specific rules apply when purging objects of type *dataspace*. These rules are described below.

The *Dataspace Maintenance Menu* is shown on page 47.

Purge Dataspace (Code *P*)

If you confirm the purge operation with DELETE, the following objects are deleted:

- the dataspace object
- all links to child objects and from parent objects
- the connection from the dataspace to the DB2 database is undone.
All DB2 tables contained in this dataspace are removed from the file list of the corresponding DB2 database object.

Dataspace Retrieval

Information on dataspace objects is retrieved with standard retrieval functions. These are described in Chapter **Retrieval** in the *Predict Reference Manual*.

Layout of Dataspace Lists

The following list format applies when retrieving information on dataspaces with the output mode *List*.

16:46:51	***** P R E D I C T 4.1.1 *****	1999-02-28
	- List Dataspace -	Page: 4

Cnt	Dataspace ID	Tablespace name
		Part Segsize
41	PD-DC2	PD_DC2
42	PD-DC3	PD_DC3
43	PD-DC4	PD_TABLE
44	PD-D1	PDPD
45	PD-TABSPACE	TABSPACE
46	* PRDSUPDB-BRUNO	BRUNO
47	* PRDSUPDB-FSTTEST	FSTTEST

Meaning of Columns

Dataspace ID	ID of the Predict dataspace object.
Tablespace ID	Name of the DB2 table space.
Part	Number of partitions.
Segsize	Size of segments.

Output Options for Dataspace Retrieval

Retrieval Type	D				B				O				T							
													dummies=Y N				dummies=D P			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																				
Adabas sizes																				
Association attributes					✓	✓	✓	✓					✓	✓	✓	✓				
Attributes	✓				✓				✓				✓				✓			
Check expression																				
Composed fields																				
Connecting Character						✓								✓						
Cover page	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Description	✓				✓	✓			✓				✓	✓			✓			
Display length																				
Display modifier	✓				✓				✓				✓				✓			
Dummy/Placeholder														✓		✓		✓		✓
DV-field expression																				
Entry points																				
Extract	✓				✓	✓			✓				✓	✓			✓	✓		
Generation layout																				
Adabas version																				
Language																				
Alignment/sync.																				
Position/Offset																				
Counter length																				
Compiler																				
Replace with syn.																				

Retrieval Type	D				B				O				T							
													dummies=Y N				dummies=D P			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓				✓	✓			✓				✓	✓			✓			
Linked verification																				
Mark implementation	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
No. abstract lines	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
Natural options																				
Owner	✓				✓	✓			✓				✓	✓			✓			
With users	✓				✓	✓			✓				✓	✓			✓			
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Procedure code																				
Rules																				
Show implementation	✓				✓				✓				✓				✓			
Sorted by field																				
Subquery																				
Synonyms																				
STARTAB elements																				
Trigger																				
Use Con-form	✓				✓	✓			✓				✓	✓			✓			
User exit	✓				✓				✓				✓				✓			
3GL specification																				

Output Options for Dataspace Retrieval (Continued)

Retrieval Type	U				E				C			
Output Mode	D		L		T		X		L		D	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes												
Adabas sizes												
Association attributes					✓	✓						
Attributes	✓					✓		✓				
Check expression												
Composed fields												
Connecting character						✓		✓				
Cover page	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Description	✓							✓				✓
Display length												
Display modifier	✓											
DV-field expression												
Dummy/Placeholder						✓		✓	✓		✓	
Entry points												
Extracts						✓		✓			✓	✓
Generation layout												
Adabas version												
Language												
Alignment/sync.												
Position/Offset												
Counter length												
Compiler												
Replace with syn.												

Retrieval Type	U				E				C			
Output Mode	D		L		T		X		L		D	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓					✓		✓				✓
Linked verification												
Mark implementation	✓		✓		✓	✓	✓	✓	✓	✓		✓
No. abstract lines	✓		✓			✓		✓		✓		✓
Natural options												
Owner	✓					✓		✓				✓
With users	✓											✓
Page size (only in batch or printout)	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Procedure code												
Rules												
Show implementation	✓											
Sorted by field												
Subquery												
Synonyms												
STARTAB elements												
Trigger												
Use Con-form	✓							✓				✓
User exit	✓											
3GL specification												

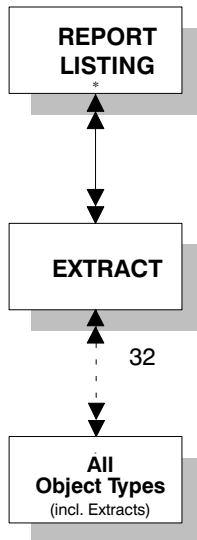
EXTRACT

An object of type *extract* in Predict fulfills two functions:

- to group objects logically
- to determine the objects to be transferred with the Coordinator.

An object can be contained in a maximum of 32 extracts. The number of objects in an extract is virtually unlimited. An extract can contain other extracts – including “itself”.

Extracts #SAG-TRANSFER and #SAG-ERROR are created automatically by the Coordinator. See the *Predict Coordinator Manual*.



In the Predict metastructure, an extract can have parents and children of the following types:

Valid Parents: Report Listing (association is created automatically)

Valid Children: *user-defined*

When you transfer objects with the Predict Coordinator, a report listing is created automatically and the extract containing the objects to be transferred is linked as a child object to this report listing.

See the *Predict Coordinator Manual*.

How this Chapter is Organized

- **Extract Maintenance**
 - The Extract Maintenance Menu, page 64
 - The Add/Copy/Modify Extract Screen, page 65
 - Extract-specific Maintenance Functions
 - Copy Extracts, page 66
 - Operate on Extracts, page 66
This function is similar to the function *Operate on sets*. An extract is increased by the result of a Union, Difference or Intersection operation.
 - Export an Extract, page 71
This function transfers data from a Predict environment to an ALF file.
 - Extract Object Editor, page 71
If you are using the SAG Editor, you can process the object list of an extract and increase the number of objects using the retrieval functionality of the SEL command.
 - Link Objects to Extract, page 77.
If you are using the Natural Editor, you can add objects of a specified type to the extract or remove objects from the extract. This function is similar to the keyword maintenance function *Link/Unlink objects*.
 - Build/extend an Extract, page 80
This function provides you with full retrieval functionality to increase the number of objects in an extract.
 - Purge Extract, page 83
- **Extract Retrieval** page 84
 - Extract-specific Retrieval Functions
 - Extracts Related to no Object, page 84
 - Extracts related to Objects, page 84
 - Output Options for Extract Retrieval, page 85

The Extract Maintenance Menu

The *Extract Maintenance Menu* is called with function code *M* and object code *ET* in a Predict main menu or with the command MAINTAIN EXTRACT.

```
10:09:17          ***** P R E D I C T  4.1.1  *****          1999-08-16
Plan    0          - (ET) Extract Maintenance -          Profile SYSTEM

Function                                Function
A  Add an Extract                      D  Display Extract
C  Copy Extract                       L  Link children
M  Modify Extract                     O  Edit owners of an Extract
N  Rename Extract                    S  Select Extract from list
P  Purge Extract                     W  Edit description
T  Operate on Extracts               E  Edit/link objects
U  Export an Extract                 B  Build/extend an Extract

Function .....
Extract ID .....
Copy ID .....

Restrictions .....*   Profile Default,empty           Child type ....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help Next Stop Last LnkEl Flip Print Impl AdmFi SelFi Prof Main
```

Parameters

Function	Standard maintenance functions are described in Chapter Maintenance in the <i>Predict Reference Manual</i> . Extract-specific maintenance functions are described starting page 66.
Extract ID	ID of the extract to be processed. See Naming Conventions , page 6.

Note:
For parameters not listed here see **Global Attributes**, page 6.

The Add/Copy/Modify Extract Screen

The following screen is displayed for functions *Add/Copy/Modify Extract*:

11:34:03

***** P R E D I C T 4.1.1 *****

1999-08-19

- Modify Extract -

Extract JCA-ET-123

Modified 1999-08-19 at 09:16

by JCA

Keys ..

Zoom: N

Abstract

Zoom: N

EDIT: Owner: N Desc: N

* Objects: N

Parameters

Note:
For parameters not listed here see **Global Attributes**, page 6.

Extract	ID of the extract.
EDIT: Objects	Y If you are using the SAG Editor, the <i>Extract Object Editor</i> is called. See page 71. If you are using the Natural Editor, the function <i>Link Objects to Extract</i> is called. See page 77.

Extract-Specific Maintenance Functions

Copy Extracts (Code C)

If you copy an extract which contains objects, an asterisk is displayed next to field *Edit: Objects*. Enter *Y* in this field. The system behavior depends on the editor you are using.

- If you are using the **SAG** Editor, the *Extract Object Editor* is called. See page 71. To copy the extract with objects, this list must be cataloged, otherwise the extract will be copied without objects.
- If you are using the **Natural** Editor, the *Link Objects to Extract* function is called. See page 77. If you confirm the object list that is displayed with ENTER, the extract is copied with objects.

Operate on Extracts (Code T)

With this function, the result of a set operation is added to the objects in the current extract (if parameter *Drop existing objects* is set to *N*) or the extract will correspond exactly to the result of the set operation (parameter *Drop existing objects*=*Y*). See overview, page 70.

Note:

An object may only be contained in a maximum of 32 extracts. If an operation would lead to one or more objects being contained in more than 32 extracts, the object(s) already contained in 32 extracts are displayed and the user has the following possibilities:

- the objects are not entered in the object list of the extract, or
- the original object list is restored.

16:46:47

***** P R E D I C T 4.1.1.1 *****

1999-04-05

- Operate on Extracts -

Extract JCA-ET1

Added 1999-02-09 at 14:04

Modified 1999-04-05 at 16:41

Operation*

Drop existing objects ... N (Y/N)

Search criteria

Extract ID

Restrictions*

Profile JCA ,empty

Parameters

Extract	ID of the extract to be processed with this operation.
Operation	<div>Enter one of the following values:</div> <div><div>U Union</div><div>You can select any number of extracts. All objects in the selected extracts are added to the current extract.</div><div>D Difference</div><div>Mark one extract with <i>X</i>, the other with <i>Y</i>.</div><div>Objects that are contained in extract <i>X</i> but not contained in extract <i>Y</i> are added to the current extract.</div><div>I Intersection</div><div>You can enter any number of extracts (but at least two).</div><div>Objects that are contained in all of the selected extracts are added to the current extract.</div><div>See overview on page 70.</div></div>
Drop existing objects	<div>Y Existing objects are removed from the object list of the extract.</div> <div>N New objects are added to existing objects in the extract.</div> <div>This parameter must be specified.</div>

Search criteria

Extract ID	<p>With this selection criterion you can limit the scope of objects to be displayed for selection.</p> <p><i>blank</i> All extracts are displayed for selection.</p> <p><i>ABC*</i> All extracts starting with ABC are displayed for selection.</p> <p>A unique extract ID makes sense only for the operation <i>Union</i>, because for <i>Difference</i> you must specify two and for <i>Intersection</i> you need at least two extracts.</p>
Restrictions	<p>Additional criteria can be selected to restrict the scope of extracts to be processed.</p>

Selecting Extracts

Enter the parameters above to display a list of extracts which meet the selection criterion *Extract ID* and any restrictions you may have entered. See example below.

16:35:03
***** P R E D I C T 4.1.1 *****
1999-08-19
Plan 11
- Select Extract from a list -

Cmd	Extract ID
—	JCA-ET-123
—	JCA-ET-124
—	JCA-ET-2
—	JCA-ET1
—	JCA-ET2
—	JCA-ET3

- For the operation **Union** you can select any number of extracts by marking them with /, X or S in the *Cmd* column.
- For the operation **Intersection** you can select any number of extracts – but at least two – by marking them with /, X or S in the *Cmd* column.
- For the operation **Difference** you must mark one extract with X and one with Y.

If you enter another command in the *Cmd* column, this command is added to the workplan. Enter an asterisk in this column to display the valid commands.

An Object can be contained in up to 32 Extracts

Objects which you want to add to the object list of the current object, but which are contained in 32 extracts already, are listed as shown in the screen below.

```
14:27:32          ***** P R E D I C T 4.1.1 *****          1999-08-2
          - Operate on Extracts -

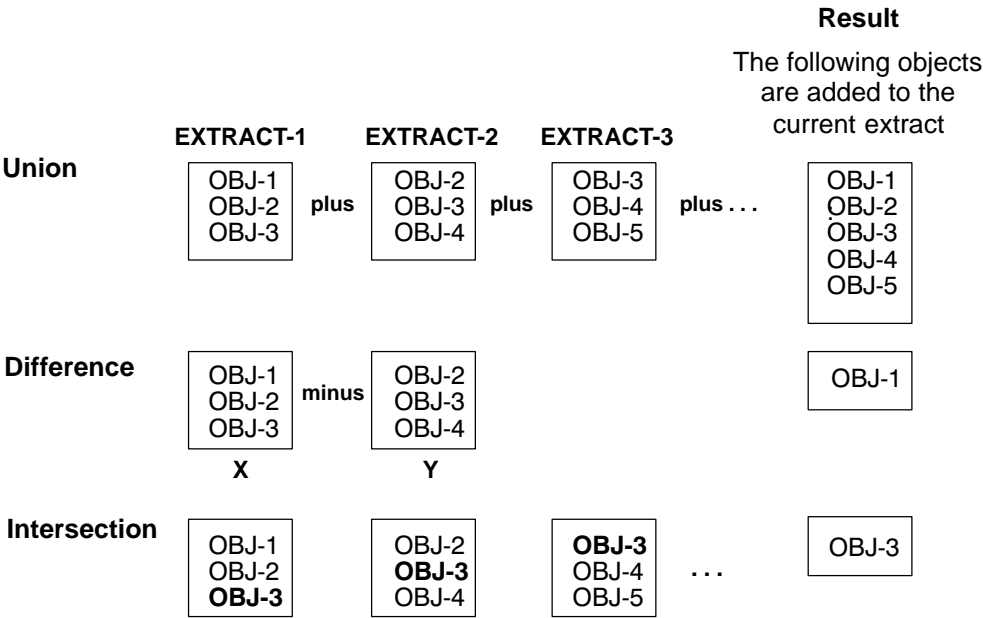
Following objects are already in 32 Extracts:

Program ..... GER-PR
Program ..... ARH-PR
```

If you confirm this list with ENTER, a window appears in which you can enter a backout option.

- | | | |
|----------------------|---|---|
| Enter backout option | Y | Terminate the operation.
No new objects are added to the current extract. |
| | N | Continue the operation.
Objects linked to fewer than 32 extracts are linked to the current extract.
Objects already linked to 32 extracts are not added to the current extract. |

Overview of Operations available for Function *Operate on Extracts*



Export an Extract (Code U)

This function transfers data from a Predict environment to an ALF file. Workfile 1 is the transfer medium. The following default parameter settings apply:

With code	N
With profile	N
With internal ID	Y
Include Extracts	N
Target environment	S

The parameters are described in detail under **Export** in Chapter **Coordinator Functions** in the *Predict Coordinator Manual*.

Edit/Link Objects (Code E)

With this function you skip the *Modify Extract* screen to edit the object list directly. The system behavior depends on which Editor is activated.

- If you are using the SAG Editor, the Object List Editor is called. See page 71.
- If you are using the Natural Editor, the function *Link Objects to Extract* is called. See page 77.

Extract Object Editor

This editor is available when you are using the SAG Editor. This editor is called

- with function code *E* from the *Extract Maintenance Menu*
- with the functions *Add/Copy/Modify Extract*: by entering *Y* in the field *EDIT Objects*
- with the command EDIT EXTRACT OBJECTS <Extract-id>.

```
10:20:26                - Extract : JCA-ET-123 -                1999-08-18
Extract object                Type Subtype
***** top of list *****
00001 JCA-BT                FI      B
00002 JCA-D2                FI      D
00003 JCA-A                 FI      A
00004 JCA-H                 DA      E
00005 JCA-D                 FI      D
00006 JCA-E                 FI      E
***** bottom of list *****
```

All functions of the SAG Editor are available. See Chapter **Editors in Predict** in the *Predict Reference Manual*.

Meaning of Columns

Extract object	ID of the object contained in the extract.
Type	Object type of the object. If you enter objects manually, you must enter ID and type.
Subtype	Subtype of the object (if applicable). If you enter ID and object type manually, the subtype is entered automatically. If an object type does not have any subtypes, this column is blank. Dummies are marked with a question mark.

Selecting Objects

- With the **SEL** command you can
- add objects of a specific type to the extract, see below
 - add objects of any type to the extract, see page 74.

With the line command **H** you can add objects to a specific position in the list.

Adding objects of a specific type

Enter the SEL command. The following screen appears in which you must enter an object type.

```
10:40:29          ***** P R E D I C T  4.1.1  *****          1999-08-18
Plan  11          - Object Selection Menu -          Profile JCA

Extract ID ..... JCA-ET-123                               Modified 1999-08-18 at 10:07
                                                    by JCA

Select object type .....*
```

The following screen appears, for example, if you specify object type *DA*.

```

08:02:31          ***** P R E D I C T  4.1.1  *****          1999-08-19
Plan  11          - Database Selection Menu -          Profile JCA

Extract ID ..... JCA-ET-123                               Modified 1999-08-18 at 10:07
                                                by JCA

Select object type ..... DA  ( Database )

Retrieval type .....* D
Output mode .....* S Select

Search criteria
  Database ID ..... Database of type*
  In Virtual machine Database number

Restrictions .....* Profile JCA ,used
                                                Related type ...*

```

Alternatively you can enter one of the following commands in the Extract Object Editor:

- SEL DA, to restrict the selection to objects of type *DA*, or
- SEL DA ABC*, to restrict the selection to objects of type *DA* which start with ABC. If only one object starts with ABC, the *Database Selection Menu* is skipped.

From this screen you can execute any retrieval function for which the output mode *Select* is valid. For Databases, for example, the following functions can be executed:

- Databases
- Dummy/Placeholder databases
- Databases with no parent
- Databases with no child

You can limit the scope of the function using selection criteria and output options. All objects which meet the selection criteria and output options are listed.

```

08:13:54          ***** P R E D I C T  4.1.1  *****          1999-08-19
Plan  11          - Select Database -

Cmd  Database ID          Type          P-DBnr  ADASTAR Parm.
---  ---
   JCA-DA1          ADABAS          122  No Translator
   JCA-H            Gen. SQL Handler    111  Local
   JCA-LEASY        Other Handler     254  Local
   JPE-10          Conceptual

```

From this list you can either

- select objects with /, S or X in the *Cmd* column to add them to the extract, or
- add functions to the workplan by entering a command other than /, S or X in the *Cmd* column. Enter an asterisk in the *Cmd* column to display the commands valid for the particular object.

Adding Objects of any type

To add objects of any type to the extract, enter one of the following commands in the Extract Object Editor:

- SEL ALL, or
- SEL, and leave the field *Select object type* in the *Object Selection Menu* empty.

The following screen appears:

```

10:38:50          ***** P R E D I C T  4.1.1  *****          1999-08-18
Plan  11          - Object Selection Menu -          Profile JCA

Extract ID ..... JCA-ET-123
                                     Modified 1999-08-18 at 10:07
                                     by JCA

Select object type ..... ( All objects )

Retrieval type .....* D
Output-mode .....* S Select

Search criteria
  Object ID .....

```

For object type *All*, only two retrieval functions are possible:

- Objects (Code *D*)
- Dummy Objects (Code *C*)

Objects of all types (except *Field* and *Owner*) are displayed for selection.

The only selection criterion is *Object ID*. With asterisk notation you can specify a range of object IDs.

You can only specify restrictions valid for all object types:

- Keywords
- Owner
- *in Extract*
- *containing the string*
- *from date*

A selection screen is displayed for each object type containing all objects that meet the selection criterion *Object ID* and any restrictions specified.

From this list you can

- select objects to be included in the extract by marking them with */*, *X* or *S* in the *Cmd* column, or
- put functions in the workplan. Enter a command other than */*, *X* or *S* in the *Cmd* column. Enter an asterisk in this column to display the commands valid for the respective object type.

All objects selected are added to the extract.

Extract-Specific Editor Commands

SORT N[AME]	Objects are sorted by columns <i>Extract object</i> and <i>Type</i> .
SORT [[T]YPE]	Objects are sorted by columns <i>Type</i> and <i>Extract object</i> .

Saving the Object List

When you have put all objects you require in the extract, enter CAT or SAVE to save the object list.

The following objects may not be added to the extract:

- duplicate objects
- non-existent objects (for example due to a typing error when adding objects manually)

If any duplicate or non-existent objects are contained in the list, the following screen appears:

```

16:25:49                - EXTRACT : JCA-ET1 -                1999-04-05
      EXTRACT OBJECT                TYPE SUBTYPE
00002  JCA-FI2                        FI

      Correct the error,
      hit      ENTER to return to the editor
      or enter  D   to remove object

DIC2519 FILE DOES NOT EXIST.
```

This screen offers you the following possibilities:

- Correct the error by overwriting an incorrect object ID or changing the type.
- Remove the object from the list by entering *D* in the prefix area on the left of the screen.
- Return to the editor by pressing ENTER.

When the list is cataloged, the extract will be added to every object in the list.

Link Objects to Extract (Code *E*)

With this function you can add objects of a specific type to the current object or remove objects from this extract. This function is available if you are using the Natural Editor and is called using one of the methods below:

- with function code *E* in the *Extract Maintenance Menu*
- with the function *Add/Copy/Modify Extract*: by entering *Y* in the *EDIT Objects* field.
- with the command EDIT EXTRACT OBJECTS <*Extract-id*>.

The following screen appears:

```
09:15:20          ***** P R E D I C T 4.1.1 *****          1999-08-18
Plan  11          - Link Objects to Extract -

Extract ID ..... JCA-ET-123                      Modified 1999-08-17 at 14:20
                                                by JCA

Link to object type ..*
```

Enter an object type to add objects of this type to the extract.

Note:

With this function you cannot enter objects of all object types in a single operation.

If you specify object type *DA*, for example, the following screen appears:

```

09:48:34          ***** P R E D I C T  4.1.1  *****          1999-08-18
Plan  11          - Link Objects to Extract -

Extract ID ..... JCA-ET-123                      Modified 1999-08-17 at 14:20
                                                by JCA

Link to object type ..* DA ( Database )

Search criteria
Database ID ..... *
Type .....*
Database number .....
in virtual machine ..

Restrictions .....*   Profile JCA ,used           List option ....* A

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Menu  Canc  S-fi  E-el  M-pr  Print Impl  Last  FLIP  PROF  Next

```

Limiting the Scope of Objects for Selection

The following possibilities are available to limit the scope of selection.

- **Search Criteria and Restrictions**

The available search criteria depend on the object type. For object type *database*, for example, you can restrict the selection with the criteria *Database ID*, *Type*, *Database number* and *in virtual machine*. You can also specify *Restrictions* to further limit the selection.

- **List Option**

This parameter determines which objects are displayed for selection.

- | | | |
|-------------|---|--|
| List option | A | All objects that meet the selection criteria and restrictions are displayed for selection. |
| | L | Only objects that meet the selection criteria and restrictions and that are linked to the current extract are displayed for selection. |
| | U | Only objects that meet the selection criteria and restrictions and that are not yet linked to the current extract are displayed for selection. |

The following screen appears for object type *DA*:

09:52:53	***** P R E D I C T 4.1.1 *****	1999-08-18
	- Link Objects to Extract -	
Extract ID	JCA-ET-123	
CMD L Database	Type	P-DBnr ADASTAR Parm.
- JCA-DA1	ADABAS	122 No Translator
- L JCA-H	Gen. SQL Handler	111 Local
- JCA-LEASY	Other Handler	254 Local

Meaning of Columns

- | | |
|-----|---|
| CMD | Enter one of the following commands: |
| | L Link the object to the current extract. |
| | U Unlink the object from the current extract. |
| L | An <i>L</i> in this column indicates that the object is already contained in the current extract. |

The other columns are type-dependent.

Build/Extend an Extract (Code B)

With this function you can create or extend the object list of an extract. The following screen is displayed:

```
10:56:01          ***** P R E D I C T  4.1.1  *****          1999-04-06
Plan    0          - Build/extend an Extract -          Profile JCA

Extract ID ..... JCA-ET1          Added 1999-04-06 at 09:54
                                   Modified 1999-04-06 at 10:02

Build Extract for object type ..*
```

Parameters

Extract ID	ID of the extract whose object list is to be added or extended.
Build extract for object type	Enter one of the following values here: <div><div><code></div><div>The two-character code of a predefined or user-defined object type. Objects of this type are added to the extract. See <i>Build/Extend an Extract for a specific object type</i>, page 81.</div></div> <div><div>blank</div><div>All object types. See <i>Build/Extend an Extract for all object types</i>, page 82.</div></div>

Build / Extend an Extract for a specified Object Type

This example shows the screen for the object type *Database*.

14:37:38
Plan 11

***** P R E D I C T 4.1.1 *****
- Build/extend an Extract -

1999-08-19
Profile JCA

Extract ID JCA-ET-123

Modified 1999-08-19 at 12:40
by JCA

Build Extract for object type ..* DA (Database)

Retrieval type*

Output mode* S Select

Search criteria

Database ID *

In Virtual machine

Database of type*
Database number

Drop existing objects N (Y,N)

List objects Y (Y,N)

Restrictions* Profile JCA ,used

Output options* Profile JCA

Model*

Related type ...*

Parameters

Retrieval type	All retrieval functions are available.
Output mode	The valid values depend on the retrieval type. Enter an asterisk to display the possible values.
Object ID	Asterisk notation is possible to specify a range of object IDs.
Search criteria	Search criteria can be used to limit the restrict the function further. These additional selection criteria are type-dependent.
Drop existing objects	Y Objects that are already contained in the extract are deleted. N New objects are added to the objects already contained in the extract.

List objects	Objects are displayed or suppressed. The default value is taken from the profile parameter <i>Maintenance options</i> > <i>List action</i> .
Restrictions	You can use <i>Restrictions</i> to further limit the scope of objects for selection. See Restrictions in Chapter Retrieval in the <i>Predict Reference Manual</i> .
Output options	With output options you can determine the amount of information displayed. See Output Options in Chapter Retrieval in the <i>Predict Reference Manual</i> . The valid output options depend on <i>Object type</i> , <i>Retrieval type</i> and <i>Output mode</i> .

Build / Extend an Extract for all Object Types

```

11:16:58          ***** P R E D I C T 4.1.1 *****          1999-04-06
Plan    0          - Build/extend an Extract -          Profile JCA

Extract ID ..... JCA-ET1                      Added 1999-04-06 at 09:54
                                                Modified 1999-04-06 at 10:02

Build Extract for object type ..*      ( All )

Retrieval type .....*
Output-mode .....*

Search criteria
  Object ID .....

Drop existing objects N (Y,N)
List objects ..... Y (Y,N)

Restrictions .....*      Profile JCA ,empty
Output options .....*      Profile JCA

```

Parameters not listed here are described above. See page 81.

Parameters

Retrieval type	With object type <i>All</i> , the following retrieval types are available: D Objects C Dummy/Placeholder Objects.
Output mode	L All objects that meet the selection criterion <i>Object ID</i> and the restrictions are put in the extract. S All objects that meet the selection criterion <i>Object ID</i> and the restrictions are displayed for selection. Enter /, X or S in the <i>Cmd</i> column to add objects to the extract. If you enter a command other than /, X or S in this column, the command is added to the workplan.
Object ID	Asterisk notation is possible. <i>Object ID</i> and <i>Restrictions</i> are the only additional selection criteria for object type <i>All</i> .
Output options	Only output options valid for all object types are displayed.
	<i>Note:</i> As you can only use output modes <i>List</i> and <i>Select</i> for this function, only the following output options are applicable: <ul style="list-style-type: none">- No. Abstract lines- Mark implementation- Cover page

Purge Extract (Code P)

This function deletes extracts and all links to other objects.

Extract Retrieval

Extract-specific Retrieval Functions

Standard retrieval functions are described in Chapter **Retrieval** in the *Predict Reference Manual*.

Extracts Related to no Object (Code Y)

Lists extracts which contain no objects.

Command:	UNUSED EXTRACT
Valid output modes	<i>Select, List, Display</i>

Extracts related to Objects (Code X)

Lists all objects contained in the current extract or – with asterisk notation – contained in a range of extracts.

Command:	XREF EXTRACT
Valid output mode:	<i>Cross reference.</i>

Layout of Extract Lists

Meaning of Columns

No. of Ref.	Number of objects contained in the extract.
-------------	---

Output Options for Extract Retrieval

Retrieval Type	D				B				O				T <i>dummies=Y/N dummies=D/P</i>							
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																				
Adabas sizes																				
Association attributes					✓	✓	✓	✓					✓	✓	✓	✓				
Attributes					✓				✓				✓				✓			
Check expression																				
Composed fields																				
Connecting character						✓								✓						
Cover page	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Description	✓				✓	✓			✓				✓	✓			✓			
Display length																				
Display modifier	✓				✓				✓				✓				✓			
Dummy/Placeholder														✓		✓		✓		✓
DV-field expression																				
Entry points																				
Extract	✓				✓	✓			✓				✓	✓			✓	✓		
Generation layout																				
Adabas version																				
Language																				
Alignment/sync.																				
Position/Offset																				
Counter length																				
Compiler																				
Replace with syn.																				

Retrieval Type	D				B				O				T <i>dummies=Y/N dummies=D/P</i>							
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓				✓	✓			✓				✓	✓			✓			
Linked verification																				
Mark implementation						✓								✓						
No. abstract lines	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
Natural options																				
Owner	✓				✓	✓			✓				✓	✓			✓			
With users	✓				✓	✓			✓				✓	✓			✓			
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Procedure code																				
Rules																				
Show implementation																				
Sorted by field																				
Subquery																				
Synonyms																				
STARTAB elements																				
Trigger																				
Use Con-form	✓				✓	✓			✓				✓	✓			✓			
User exit	✓				✓				✓				✓				✓			
3GL specification																				

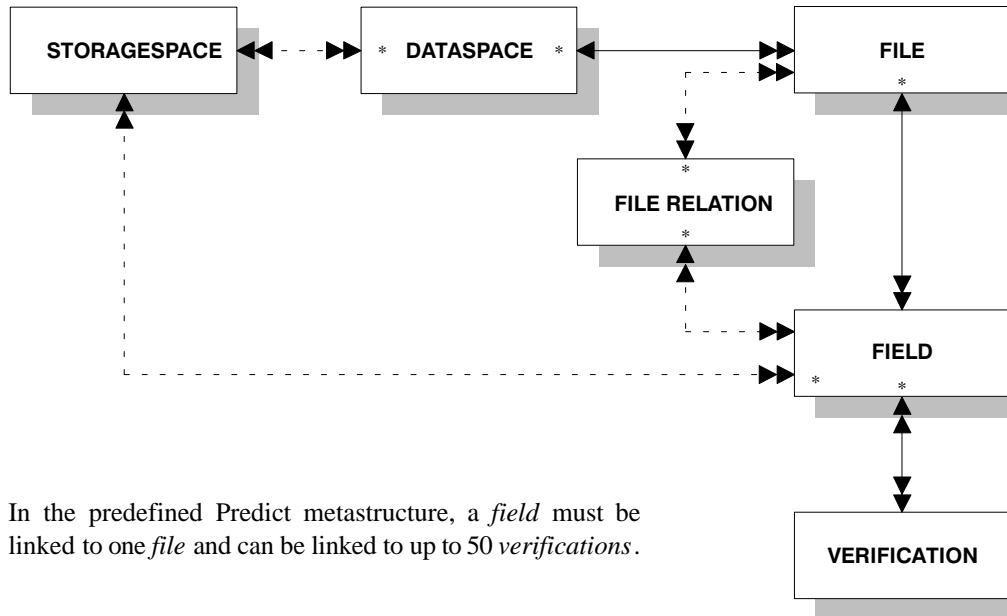
Output Options for Extract Retrieval (Continued)

Retrieval Type	U				E				C				Y				X			
Output Mode	D		L		T		X		L		D		D		L		X			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																				
Adabas sizes																				
Association attributes					✓	✓														
Attributes						✓		✓										✓		
Check expression																				
Composed fields																				
Connecting character						✓		✓										✓		
Cover page	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		
Description	✓							✓				✓	✓					✓		
Display length																				
Display modifier	✓												✓							
Dummy/Placeholder						✓		✓	✓		✓									
DV-field expression																				
Entry points																				
Extract	✓					✓		✓			✓	✓	✓					✓		
Generation layout																				
Adabas version																				
Language																				
Alignment/sync.																				
Position/Offset																				
Counter length																				
Compiler																				
Replace with syn.																				

Retrieval Type	U				E				C				Y				X			
Output Mode	D		L		T		X		L		D		D		L		X			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓					✓		✓				✓	✓					✓		
Linked verification																				
Mark implementation						✓		✓		✓		✓						✓		
No. abstract lines	✓		✓			✓		✓		✓		✓	✓		✓			✓		
Natural options																				
Owner	✓					✓		✓				✓	✓					✓		
With users	✓											✓	✓					✓		
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		
Procedure code																				
Rules																				
Show implementation																				
Sorted by field																				
Subquery																				
Synonyms																				
STARTAB elements																				
Trigger																				
Use Con-form	✓							✓				✓	✓					✓		
User exit	✓												✓							
3GL specification																				

FIELD

With Predict, data definitions can be documented for a wide variety of data storage systems and for use with different programming languages. Field definitions are documented with objects of type *field*.



In the predefined Predict metastructure, a *field* must be linked to one *file* and can be linked to up to 50 *verifications*.

How this Chapter is Organized

- **The Field Maintenance Menu** page 93
- **Defining Basic Attributes of Fields** page 95
This section describes general attributes. Most of these are applicable to fields of all file types. Type-specific attributes are described in later sections.
- **Defining Derived Fields** page 122
A derived field is a term used in Predict for fields and descriptors defined on the basis of one or more source fields. This section tells you how to define the different types of derived fields.
- **Defining More Attributes of Fields** page 139
 - 3GL Specification, page 140
 - Condition Name & Value, page 144
 - Field Name Synonyms, page 145
 - Old Mode Synonyms, page 142
 - Adabas Security & Edit mask, page 146
 - Field Procedure, page 147
 - Derived Field Expression, page 148
 - Index Definition (DB2), page 149
 - Default name, page 153
 - Constraint name, page 154
- **Field Maintenance** page 155
 - Add a Field, page 156
 - Copy Field, page 156
 - Move Field within a File, page 157
 - Purge Field, page 157
 - Redefine Field, page 158
 - Browse through Fields of a File, page 163
 - Link Verification, page 163
 - Edit Field expression, page 163

- **Field Retrieval** page 164
 - Field Specific Retrieval Parameters, page 164
 - Sorting Files and Fields, page 166
 - Field Specific Retrieval Functions, page 168
 - Layout of Field Lists, page 170
 - Output Options, page 172

The Field Maintenance Menu

The *Field Maintenance Menu* is called with function code *M* and object code *EL* in a Predict main menu or with the command MAINTAIN ELEMENT.

The functions *Add a Field* and *Modify Field* can also be called with the editor line command *E* when maintaining the field list of a file object.

```

12:26:29          ***** P R E D I C T  4.1.1  *****          1999-02-25
Plan    0          - (EL) Field Maintenance -          Profile: JCA

Function                                Function

A  Add a field                          B  Browse through fields of a file
C  Copy field                          H  Move field within a file
D  Display field                       L  Link verification
M  Modify field                       O  Edit owners of a field
N  Rename field                      S  Select field from a list
P  Purge field                       W  Edit description of a field
R  Redefine field                    Y  Edit field expression

Function .....
Field ID .....
in file .....          in file of type ....*
Copy field ID ....
Copy file ID .....
Restrictions ....*   Profile JCA ,used

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Menu  Canc  S-fi  E-el  M-pr  Print Impl  Last  FLIP  PROF  Next

```

Parameters

Function	<p>Executes one of the maintenance functions.</p> <p>The following functions are described in this chapter:</p> <ul style="list-style-type: none"> - <i>Add a Field</i>, page 156 - <i>Copy Field</i>, page 156 - <i>Move Field within a File</i>, page 157 - <i>Purge Field</i>, page 157 - <i>Redefine Field</i>, page 158 - <i>Browse through Fields of a File</i>, page 163 - <i>Link Verification</i>, page 163 - <i>Edit Field expression</i>, page 163 <p>All other functions are described in Chapter Maintenance in the <i>Predict Reference Manual</i>.</p>
Field ID	<p>See Naming Conventions, page 6.</p> <p>For the <i>Select</i> function:</p> <p>specifies a field ID which is to be used as a selection criterion. The field ID can be used alone or in combination with the file id. if this field is left blank, all fields in the specified file(s) are listed.</p> <p>Asterisk notation is possible.</p>
Copy Field ID	<p>Specifies the ID of a field that is added or the position of a field that is copied or moved. See Copy Field, page 156.</p> <p>For functions <i>Add a Field</i> and <i>Move Field within a File</i>: the position of the newly added or moved field. See page 156 and page 157 respectively.</p>
in File	<p>For the <i>Add/Copy/Modify</i> function: file containing the field.</p> <p>For the <i>Select Field</i> function: File ID is used as a selection criterion, either alone or in combination with the field ID.</p> <p>Asterisk notation is possible. If this field is left blank, all files are included in the search.</p>
in File of type	<p>For the <i>Select Field</i> function:</p> <p>The scope of the function is restricted to fields in files of the specified type.</p>
Copy File ID	<p>Used for function <i>Copy Field</i> to identify the file to which a field is copied. See Copy Field, page 156.</p>

Defining Basic Attributes of Fields

The functions *Add Field* and *Modify Field* can also be called from within the function *Edit elements of a File* with the editor line command *.E*. See Chapter **Editors in Predict** in the *Predict Reference Manual*.

How this Section is Organized

This section describes the following general attributes. Most attributes are applicable to fields of all file types. Type-specific attributes are described in later sections.

- *Add / Copy / Modify Field* Screen (for Fields of non-SQL Files), page 96
- *Add / Copy / Modify* Screen for SQL Fields, page 97
- Field Type, page 98
- Level Number, page 99
- Field Format, page 100
- Character Set (only for SQL File types), page 101
- Field Length, page 102
- DBMS Format (only for SQL File types), page 109
- Descriptor Type, page 113
- Maximum Number of Values / Occurrences, page 115
- Unique Option, page 115
- Field Short Name, page 116
- Suppression / Null Value Option, page 117
- Variable Length Option (IMS), page 118
- Natural Field Length, page 119
- Null Default Option, page 118
- Related standard File, page 119
- Check against standard, page 119
- Natural Attributes, page 120
- EDIT Line Options, page 121

Basic attributes applying to different field types are described below. Type-specific attributes are described in later sections.

Add / Copy / Modify Screen

The following screen is displayed for the functions *Add/Copy/Modify Field*.

12:01:02

***** P R E D I C T 4.1.1 *****

1999-02-25

- Add a Field -

Field ID JCA-EL-NEW

File ID JCA-FI1

Keys ..

Zoom: N

Ty	L	Field ID	F	Length	Occ	D	U	DB	N	NAT-1
*	-	-----	*	-----	-----	*	*	--	*	-----
1		JCA-EL-NEW							AC	N

NATURAL attributes

Header1

Header2

Header3

Edit mask ..

Abstract

Zoom: N

EDIT: Owner: N Desc: N Veri: N MORE Attr.: N

The screen for maintaining fields of SQL file types contains some different attributes and is shown below.

Add / Copy / Modify Screen for SQL Fields

The following screen is displayed for the functions *Add/Copy/Modify Field* and applies to SQL file types.

- A Adabas C (with parameter *Adabas C SQL usage* set to *Y*)
- AT Adabas Cluster Table
- B Adabas C SQL view
- BT, BV Adabas D table/view
- D, E DB2 table/view
- OT, OV ORACLE table/view
- JT, JV INGRES table/view
- X General SQL file
- XT, XV INFORMIX table/view
- YT, YV SYBASE table/view

10:06:11

***** P R E D I C T 4.1.1 *****

1999-06-22

- Add a Field -

Field ID DB1

File ID JCA-DB2

Keys ..

Zoom: N

Ty	L	Field ID	F	Cs	Length	Occ	D	U	N	Df	NAT-1
*	-	-----	*	*	-----	-----	*	*	*	*	-----
1		DB1									U

NATURAL attributes

Header1

Header2

Header3

Edit mask ..

Comments

Zoom: N

EDIT: Owner: N Desc: N Veri: N

MORE Attr.: N

Field Type

The field type is indicated in the column *Ty* of the *Add a field* screen on page 96.

The following types can be specified:

CM	Counter Field for multiple value field of type MU/MC
CP	Counter Field for periodic group of type PE/PC
DV	Derived field (SQL File types) see note below
GR	Group
HM	Hyperdescriptor as a multiple value field
HP	Hyperdescriptor as a field of a periodic group
HQ	Hyperdescriptor as a multiple value field of a periodic group
HY	Hyperdescriptor
MC	Multiple value field with automatic counter
MU	Multiple value field
PC	Periodic group with automatic counter
PE	Periodic group
PH	Phonetic descriptor
QN	SEQNO field
SB	Subfield/descriptor
SP	Superfield/descriptor
**, /*	Comment line, see page 259
blank	None of the above. Normal field

Note:

Derived field is also used in Predict as a generic term for hyperdescriptors, phonetic descriptors and sub/superfields and descriptors.

If *HM*, *HP*, *HQ*, *HY*, *PH*, *SB* or *SP* is specified, an additional input screen is displayed. See **Defining Derived Fields**, page 122.

See also Chapter **ADACMP (COMPRESS- DECOMPRESS)** in the *Adabas Utilities Manual*.

Redefining Fields

See **Redefine Fields**, page 158.

Defining Periodic Groups in Periodic Groups

- Within a redefinition, nested periodic groups (PE within a PE) can be defined in files of all types.
- Outside of a redefinition, nested periodic groups can only be defined in files of the following types:

S	Sequential file
C	Conceptual file
M	ISAM file
Z	Standard file
O	Other file

Level Number

The level number of the field is indicated in the column *L* of the *Add a field* screen on page 96. The level number is used to define a group structure. Level numbers 1 to 9 can be used (except for Adabas files, see below).

- The level number must be increment by 1 immediately following a field of type *RE*, *PE*, *PC* or *GR*.
- For redefinitions, the level number must be at least one greater than the level number of the field being redefined. See **Redefine Field**, page 158.

Adabas Files

The following rules apply to level numbers for Adabas files:

- The PE/PC-groups, sub/superfields/descriptors, hyperdescriptors and phonetic descriptors must be at level 1.
- Level numbers of fields outside a redefinition must be in the range 1 - 7. See Chapter **ADACMP (COMPRESS- DECOMPRESS)** of the *Adabas Utilities Manual* for a complete description of Adabas levels.

Field Format

The format of the field is indicated in the column *F* of the *Add a field* screen on page 96. One of the following values can be specified (depending on the file type):

A	Alphanumeric	L	Logical
AL	Long varchar	LO	Large object
AV	Varchar	LX	Bfile
B	Binary/char for bit data	MO	Money
BL	Long varchar bit data	MS	Smallmoney
BT	Bit	N/U	Numeric unpacked
BV	Varchar for bit data	NS/US	Numeric unpacked with sign
D	Date	OK	Object key
DS	Smalldatetime	P	Packed numeric
DT	Datetime	PS	Packed numeric with sign
F	Floating point	S	Serial
G	Graphic	T	Time
GL	Long vargraphic	TK	Table key
GV	Vargraphic (DB2)	TS	Time stamp
I	Integer	<i>blank</i>	Undefined
IV	Interval		

See table on page 105 for valid combinations of format and length.

The following rules apply:

- Any format/length combination is allowed for the file types C (conceptual) or Z (standard).
- For groups, this attribute must be blank.
- For sub/superfields/descriptors in Adabas files, the appropriate format is provided by Predict based on the formats of the fields used. See **Rules Applying to Format Changes**, page 128.
- The following formats are valid for **Fields** within a definition: A, B, D, F, I, L, N/U, NS/US, P, PS, T

Character Set (SQL)

The parameter *Character set* determines the format in which data is stored. It is indicated in column *Cs* of the *Add a field* screen on page 97. This attribute applies only to fields in SQL files. The possible values depend on the file type and format.

File Type	Format	Character Set					
		ASCII	EBCDIC	Bitdata	Single Byte	Double Byte	Mixed Data
Adabas D	A, AL, AV	✓	✓	✓			
DB2	A, AL, AV			✓	✓		✓
ORACLE	A, AL			✓			
	AV						✓
	LO			✓			✓
INFORMIX	A, AV						✓
	AL			✓			
INGRES	A,AV			✓			
	AL *			✓			
SYBASE	A, AV *			✓	✓	✓	
	AL			✓			

Note:

A value must be specified for field types and formats marked with an asterisk (*).

Character Set

ASCII	Data is stored in ASCII format.
EBCDIC	Data is stored in EBCDIC format.
Bitdata	Data is stored in binary form, no conversion is performed.
Single Byte	Data is stored in single-byte format. Double-byte characters are not possible.
Double Byte	Data is stored in double-byte format. String comparisons function differently to single-byte data.
Mixed Data	Data is stored in single and double-bytes. Data is subject to DB2 rules for multiple-byte character sets.

Field Length

The field length is indicated in column *Length* of the *Add a field* screen on page 96. This length is independent of its internal representation. When generating external objects, the field length is adjusted according to the internal representation of data used by the data storage system. For example: a field which is documented with length P9 is implemented with length P5 by the Adabas LOADER utility and the Adabas nucleus.

See table on page 105 for valid combinations of format and length.

The following additional rules apply:

- For files of type *C* (conceptual) or *Z* (standard):
Any format/length combination is allowed, and field length zero is permitted for all field formats.
- For groups and phonetic descriptors:
Field length must be set to zero.
- For sub/superfields/descriptors in Adabas files:
The appropriate length is provided by Predict based on the definition.

Table of Field Formats and Lengths

The table on the following pages contains the valid format/length combinations for fields of the following file types:

Column	File Type
A / U	Adabas C file / userview
A(SQL) / AT / B	Adabas C file with <i>SQL usage</i> , Adabas table cluster, Adabas C SQL view
BT / BV	Adabas D table / view
D / E	DB2 table / view
F	rdb file
I / J / K	IMS segment / segment layout / userview
JT / JV	INGRES table / view
L / R / V / W	Logical VSAM file / view / Physical VSAM file / view
M	ISAM file
O	Other
OT / OV	ORACLE table / view
P / Q	Entire System Server file / userview
S	Sequential file
T	RMS file
X	General SQL file
XT / XV	INFORMIX table / view
YT / YV	SYBASE table / view
1	LEASY
2	ISAM BS2000

Note:
The tables do not contain the file types C (conceptual) and Z (standard). For these file types, any format/length combinations are allowed.

Key

no length	Format is valid; length must not be specified.
no restr.	No restrictions: any length may be specified.
p.q (m / n)	p: number of places before the decimal point q: number of places after the decimal point where 0 <= p <= m 0 <= q <= n 1 <= p+q <= m
n.m – n2.m2	Range of places before and after the decimal point. For example, fields of format <i>MO</i> for SYBASE tables and views can have up to 15 places before the decimal point and up to 4 places after the decimal point (1.0 – 15.04).
*1	0 means 2GB
*2	0 means 4GB

Field Format	A , U	A(SQL) AT, B	BT, BV	D , E	F	I , J , K	JT , JV	L ,R, V, W	M
A	1-253	1-253	1-4000	1-254	1-253	1-253	1-2000	1-253	no restr.
AL			0-99999 *1	1-99999			0-99999 *1		
AV	1-16381	1-16381	1-4000	1-32767			1-2000		
B	1-126	1-126			1-126	1-126	1-2000	1-126	no restr.
BL							0-99999 *1		
BT									
BV							1-2000		
C									
D	no length	no length	no length	no length	no length	no length		no length	no length
DS									
DT							no length		
F	4 / 8	4 / 8	4 / 8	4 / 8		4 / 8	4 / 8	4 / 8	4 / 8
G				1-127					
GL				1-16383					
GV				1-16383					
I	1 / 2 / 4 / 8	1 / 2 / 4 / 8	2 / 4	2 / 4	1 / 2 / 4 / 8	1 / 2 / 4 / 8	1 / 2 / 4	1 / 2 / 4 / 8	1 / 2 / 4 / 8
IV									
L	no length		no length			no length		no length	no length
LO									
LX									
MO							no length		
MS									
N	p.q (29/7)	p.q (29/7)			p.q(29/29)	p.q (29 / 7)		p.q (29 / 7)	p.q(29/29)
NS	p.q (29 / 7)	p.q (29 / 7)	p.q(18/18)	p.q(31/31)	p.q(29/29)	p.q (29 / 7)		p.q (29 / 7)	p.q(29/29)
OK							no length		
P	p.q (29 / 7)	p.q (29 / 7)			p.q(29/29)	p.q (29 / 7)		p.q (29 / 7)	p.q(29/29)
PS	p.q (29 / 7)	p.q (29 / 7)	p.q(18/18)	p.q(31/31)	p.q(29/29)	p.q (29 / 7)		p.q (29 / 7)	p.q(29/29)

O	OT, OV	P,Q	S	T	X	XT, XV	YT, YV	1	2
no restr.	1-2000	no restr.	no restr.	1-253	1-253	1-32762	1-255	1-253	1-253
	0-99999 *1					0-99999 *1	0-99999 *1		
	1-2000					1-32762	1-255		
no restr.		1-126	no restr.	1-126			1	1-126	1-126
							no length		
no length		no length	no length	no length		no length		no length	no length
							no length		
	no length					no length	no length		
4 / 8	4 / 8	4 / 8	4 / 8		4 / 8	4 / 8	4 / 8	4 / 8	4 / 8
1 / 2 / 4 / 8	2 / 4	1 / 2 / 4 / 8	1 / 2 / 4 / 8	1 / 2 / 4 / 8	1 / 2 / 4 / 8	2 / 4	1 / 2 / 4	1 / 2 / 4 / 8	1 / 2 / 4 / 8
						7 / 17			
no length		no length	no length					no length	no length
	0-99999 *2								
	0-99999 *2								
						p.q(32767/99)	1.0 - 15.04		
							1.0 - 6.04		
p.q(29/29)		p.q(29/29)	p.q(29/29)	p.q(29/29)	p.q(32/32)		p.q(38/38)	p.q(29/29)	p.q(29/29)
p.q(29/29)	p.q(29/29)	p.q(29/29)	p.q(29/29)	p.q(29/29)		p.q(32 32)		p.q(29/29)	p.q(29/29)
p.q(29/29)		p.q(29/29)	p.q(29/29)	p.q(29/29)	p.q(32/32)			p.q(29/29)	p.q(29/29)
p.q(29/29)	p.q(29/29)	p.q(29/29)	p.q(29/29)	p.q(29/29)		p.q(32/32)		p.q(29/29)	p.q(29/29)

Field Format	A , U	A(SQL) AT, B	BT, BV	D , E	F	I , J , K	JT , JV	L ,R, V, W	M
S									
T	no length	no length	no length	no length	no length	no length		no length	no length
TK							no length		
TS			no length	no length					
U	p.q (29 / 7)	p.q (29 / 7)			p.q(29/29)	p.q (29 / 7)		p.q (29 / 7)	p.q(29/29)
US	p.q (29 / 7)	p.q (29 / 7)	p.q(18/18)	p.q(31/31)	p.q(29/29)	p.q (29 / 7)		p.q (29 / 7)	p.q(29/29)

O	OT, OV	P,Q	S	T	X	XT, XV	YT, YV	1	2
						no length			
no length		no length	no length	no length				no length	no length
							no length		
p.q(29/29)		p.q(29/29)	p.q(29/29)	p.q(29/29)	p.q(32/32)			p.q(29/29)	p.q(29/29)
p.q(29/29)	p.q(29/29)	p.q(29/29)	p.q(29/29)	p.q(29/29)		p.q(32/32)		p.q(29/29)	p.q(29/29)

SQL: DBMS Format and Corresponding Predict Format

The table below indicates the DBMS format and the corresponding Predict format for fields in files of the following types:

BT, BV	Adabas D table/view
JT, JV	INGRES table/view
OT, OV	ORACLE table/view
XT, XV	INFORMIX table/view
YT, YV	SYBASE table/view

Key

n	length
p,q	p: total number of places
	q: number of places after the decimal point

File Type	DBMS Format	Predict Format	Character Set
BT, BV	BOOLEAN	L	ASCII Bitdata EBCDIC
	CHAR(n)	A(n)	
	CHAR(n) ASCII	A(n)	
	CHAR(n) BYTE	A(n)	
	CHAR(n) EBCDIC	A(n)	
	DATE	D	
	FIXED(p,q)	NU, US, or PS	
	FLOAT(15)	F4	
	FLOAT(18)	F8	
	INTEGER	I4	
	LONG	AL	ASCII Bitdata EBCDIC
	LONG ASCII	AL	
	LONG BYTE	AL	
	LONG EBCDIC	AL	
	SMALLINT	I2	
	TIME	T	ASCII Bitdata EBCDIC
	TIMESTAMP	TS	
	VARCHAR(n)	AV(n)	
	VARCHAR(n) ASCII	AV(n)	
	VARCHAR(n) BYTE	AV(n)	
	VARCHAR(n) EBCDIC	AV(n)	
JT, JV	BYTE(n)	B	Bitdata
	BYTE VARYING	BV	
	C(n)	A(n)	
	CHAR(n)	A(n)	
	DATE	DT	
	DECIMAL (p,q)	PS	
	DECIMAL (p,q)	NS	
	DOUBLE PRECISION	F8	
	LONG BYTE	BL	Bitdata
	LONG VARCHAR	AL	
	INTEGER	I4	
	INTEGER1	I1	
	MONEY	MO	
	OBJECT_KEY	OK	
	REAL	F4	
	SMALLINT	I2	
	TABLE_KEY	TK	

File Type	DBMS Format	Predict Format	Character Set
	TEXT(n) VARCHAR(n)	AV(n) AV(n)	Bitdata
OT, OV	BFILE BLOB CHAR(n) CLOB DATE DECIMAL(p,q) DECIMAL(p,q) DOUBLE PRECISION INTEGER LONG LONG RAW NCLOB NVARCHAR2(n) RAW(n) REAL ROWID SMALLINT VARCHAR2(n)	LX LO A(n) LO DT NS PS F8 I4 AL AL LO AV(n) A(n) F4 A and type QN I2 AV(n)	Bitdata Bitdata Mixed data Mixed data Bitdata
XT, XV	BYTE CHAR(n) DATE DATETIME YEAR TO FRACTION(5) DECIMAL(p,q) DECIMAL(p,q) FLOAT INTEGER INTERVAL DAY TO FRACTION(5) MONEY NCHAR(n) NVARCHAR(n) REAL SERIAL SMALLINT TEXT VARCHAR(n)	AL A(n) D DT NS PS F8 I4 IV MO A(n) AV(n) F4 S I2 AL AV(n)	Bitdata Mixed data Mixed data

File Type	DBMS Format	Predict Format	Character Set
YT, YV	BINARY(N)	A(n)	Bitdata
	BIT	BT	
	CHAR(N)	A(n)	Single byte
	DATETIME	DT	
	FLOAT	F8	
	IMAGE	AL	Bitdata
	INT	I4	
	MONEY	MO	
	NCHAR(N)	A(n)	Double byte
	NUMERIC, DECIMAL (p,q)	NS	
	NUMERIC, DECIMAL (p,q)	PS	
	NVARCHAR(N)	AV(n)	Double byte
	REAL	F4	
	SMALLDATETIME	DS	
	SMALLINT	I2	
	SMALLMONEY	MS	
	TEXT	AL	
	TIMESTAMP	TS	
	TINYINT	I1 or B1	
	VARBINARY(N)	AV(n)	Bitdata
	VARCHAR(N)	AV(n)	Single byte

Descriptor Type

The descriptor type is indicated in column *D* of the *Add a field* screen on page 96. The possible values are given in the table on this and the opposite page.

Code	Description	File Type								
		A,U	AT,B, A(SQL)	M	O	F	S	T	D,E	C
D	Descriptor/Index	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Disallow									
A	Alternate index									✓
N	Not inverted	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Search field									
P	Primary Index		✓						✓	✓
Q	Sequence									✓
E	Foreign key		✓						✓	✓
F	Foreign index		✓						✓	✓
	Force									
K	Common Key									
blank	No descriptor	✓	✓	✓	✓	✓	✓	✓	✓	✓
	None									

The following rules apply:

- In an Adabas C file, the descriptor must be *D* if type *HM*, *HP*, *HQ*, *HY* (all hyperdescriptors) or *PH* (phonetic descriptor) is specified.
- For a subdescriptor in an Adabas C file, descriptor *D* and type *SB* (subfield) must be specified.
- For a superdescriptor in an Adabas C file, descriptor *D* and type *SP* (superfield) must be specified.
- In a DB2 table, if a key or index (descriptor *D*, *E*, *F* or *P*) includes more than one field, the type *SP* (superfield) must be specified.

Descriptor Type (continued)

File Type											
I,J,K	P,Q	Z	1	2	L,R,V,W	X	BT,BV	OT,OV	JT,JV	YT,YV	XT,XV
	✓	✓									
✓			✓	✓	✓						
✓						✓	✓	✓	✓	✓	✓
			✓	✓	✓	✓	✓	✓	✓	✓	✓
✓											
						✓	✓	✓	✓	✓	✓
		✓									
										✓	
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

- In a VSAM file or userview (type *L*, *R*, *V* or *W*), the descriptor must be either *P* or *A* if type *SP* (superfield) is specified.
- If *A* is specified for a field of a VSAM file (type *L* or *V*), an additional screen is displayed for entering the required definitions (see below).
- Descriptor type must be *blank* for fields within a redefinition.

Maximum Number of Values / Occurrences

Maximum number of values for a multiple value field or occurrences of a periodic group is indicated in the *Occ* column of the *Add a field* screen on page 96. This parameter must be specified for multiple value fields and for periodic groups in a redefinition.

Field	Occurrences in range
Within a redefinition	1– 99999
Outside a redefinition	1 – 191

When generating Copy Code, the value specified is used as the default for generating the specifications of *MU/MC* or *PE/PC* fields in a format buffer and/or record buffer.

When generating ADAWAN/ADACMP definitions, ADAWAN/ADACMP evaluates the *Occ* parameter. If *Occ* is specified, the number of occurrences of each input data record is constant.

If *Occ* is not specified, the number of occurrences is taken from a counter field preceding a *MU/MC* or *PE/PC* field.

See also the Chapter **ADACMP (COMPRESS–DECOMPRESS)** in the *Adabas Utilities Manual*.

Note:
For fields of type *QN*, the *Occ* column is used to identify either the table level or an individual occurrence of a multiple value field or periodic group.

Unique Option

The unique option is indicated in column *U* of the *Add a field* screen on page 96. For groups, this attribute must be blank; for other fields, one of the following values can be specified:

U Unique
blank Not unique

Unique option must be *blank* for fields within a redefinition.

Field Short Name

For file types listed below, the field short name is indicated in the column *DB* of the *Add a field* screen on page 96. This two-character short name must be defined for the following file types:

A	Adabas C file	L	Logical VSAM file
AT	Adabas cluster table	R	Logical VSAM view
I	IMS segment	U	Adabas C userview
J	IMS segment layout	V	VSAM file (physical)
K	IMS userview	W	Physical VSAM view

A field short name must conform to the rules for coding Adabas field names. See chapter **ADACMP (COMPRESS-DECOMPRESS)** in the *Adabas Utilities Manual*.

Field short names for userviews of Adabas, IMS and VSAM files need not be unique.

For fields within a redefinition, parameter *Field short name* must be blank.

Field short names for SQL tables and views are maintained internally by Predict and cannot be modified by users.

Rotated fields of files of type *A* (with *SQL usage*), type *AT* and *B* have the same short name and are identified uniquely by an occurrence number (column *Occ*).

Suppression / Null Value Option

- For fields of Adabas C files, the suppression option is indicated in column *S* of the *Add a field* screen on page 96.
- For fields of SQL files, the null value option is indicated in column *N* of the *Add a field* screen on page 97.

For groups and for fields within a redefinition, this attribute must be blank. For other fields, one of the following values can be specified:

- F Fixed length
- N Null value suppression
- R Not null
- U Null counted
- blank* Normal suppression

Parameter	SQL File Types *	Other File Types
Null value suppression		N
Fixed Length		F
Null allowed	U	U
Not null	R	R
Normal suppression		<i>blank</i>

* See page 97 for a list of SQL file types.

See also chapter **ADACMP (COMPRESS-DECOMPRESS)** in the *Adabas Utilities Manual*.

Profile Parameter *Automatic Null Value*

With the profile parameter *Automatic null value* you can determine an automatic *Suppression/Null Value* option when fields are added in Predict. See **Customizing Predict with Profiles** in Chapter **The User Interface** in the Manual *Introduction to Predict*.

The value depends on the file type. See table below.

Parameter	All SQL File Types * except <i>X</i>	File Type <i>X</i>	Other File Types
Unique option = <i>Unique</i> or Descriptor type = <i>Primary</i> or Field format = <i>serial</i>	R	R	N
Others	U	blank	N

Note:

SQL file types include files of type *A* with parameter *Adabas C SQL usage* set to *Y*. See list on page 97.

Note:

For DB2 fields with *Unique option* = *unique*, values *R* and *U* are possible.

Variable Length Option (IMS)

The variable length option for IMS fields is indicated in column *S* of the *Add a field* screen on page 96. The following values are valid:

Y Variable length
blank Fixed length.

Null Default Option

The NULL default option for fields of SQL tables/views is indicated in the *Df* column of the *Add a field* screen on page 97. Possible values:

N No default
Y With default
blank none

For INGRES fields with format *OK* or *TK*, the following additional values are possible:

S	SYSTEM_MAINTAINED
T	not SYSTEM_MAINTAINED
U	with default SYSTEM_MAINTAINED
V	with default not SYSTEM_MAINTAINED
W	not default not SYSTEM_MAINTAINED

This parameter must be *blank* for fields within a redefinition.

Natural Field Length

The Natural field length is indicated in column *NAT-l* of the *Add a field* screen on page 96. The following rules apply:

- The parameter has to be specified if the field can be:
 - alphanumeric and greater than 253
 - graphic and greater than 126
 - numeric *p.q (m/n)* where $p+q > 29$ or $q > 7$.

See table of valid formats and lengths on page 105.

- The value specified here is the length that Natural uses for the field as defined in the DDM.

Related Standard File

If the field is connected to a corresponding field in a standard file, Predict places the name of the standard file in this field. This attribute is only displayed for fields in files that are connected to standard files.

Check against standard

This parameter determines the handling of fields connected to standard fields. If *N* (non-standard) is specified, fields are not checked against the definition of the standard field from which they have been derived. In this case, fields can be modified independently of the standard field, and modifications made to attributes of the standard field are not rippled to the field. This attribute is only displayed for fields in files that are connected to standard files.

Note:

Even with *Check against standard* set to *N*, a derived field and the standard field remain coupled and a change of the name of a standard field is still rippled to a derived field if they are identical. The option *D* can be used to purge the connection of a field to the standard file.

This parameter is also described in section **Rippling**, page 264, in Chapter **File**.

Natural Attributes

Natural Header 1 – 3

Natural Header1	The first line header is used for the field in reports and for labels when generating SQL tables/views.
Natural Header2	The second line header to be used for the field in reports.
Natural Header3	The third line header to be used for the field in reports.

The Natural headers 1 - 3 are included in DDMs generated from the file containing the field.

Alphabetic characters in Natural headers are converted to upper-case if the Predict parameter *Upper/lower case* has been set to *Y*. See Chapter **Defaults** in the *Predict Administration Manual*.

Index on PE Group Level

If you enter *Y* in this field and execute the Natural Area Editor command *.V* for a DDM containing the field object, the maximum occurrences of periodic groups is generated on group level.

If this parameter is left blank (default), the maximum occurrences is generated for each element in the group.

Edit mask

The Natural edit mask. See the description of the DISPLAY statement in the *Natural 2 Reference Manual* for further details.

Alphabetic characters in the Natural edit mask are converted to upper-case if the Predict parameter *Upper/lower case* has been set to *Y*. See Chapter **Defaults** in the *Predict Administration Manual*.

EDIT Line Options

The following additional EDIT line options are available for fields. standard options are described on page 10.

EDIT Veri.

Enter *Y* in this field to call the Predict Link Editor to edit the verification list of the field. Up to 50 verifications can be linked to a field.

This editor can also be invoked by:

- Selecting *L* (link verification) in the *Field Maintenance Menu*.
- Entering command LINK ELEMENT VERIFICATION *Field ID File ID*.

See Chapter **Editors in Predict** in the *Predict Reference Manual*.

MORE Attr.

Enter *Y* in this field to define additional attributes. The attributes that can be defined depend on the field type. See **Defining More Attributes of Fields** on page 139.

- 3GL Specification, page 140
- Condition Name & Value, page 144
- Field Name Synonyms, page 145
- Old Mode Synonyms, page 142
- Adabas Security & Edit mask, page 146
- Field Procedure, page 147
- Derived Field Expression, page 148
- Defining a DB2 Index, page 149
- Default name, page 153
- Constraint name, page 154

Defining Derived Fields

Note:

Derived field is a generic term in Predict for fields and descriptors defined on the basis of one or more source fields. This term should not be confused with field type *DV* applicable to SQL views (see page 98).

Defining derived fields and keeping the definitions consistent is a complex task. Predict offers a variety of functions to help with it. General rules applying to the definition of derived fields are described in the following sections:

- **General Rules for Defining Derived Fields** page 123
 - Specifying the start and end position
 - Editor functions
 - Selection mechanism
- **Rules Applying to Format Changes** page 128

The format of derived fields is determined by Predict or can be defined manually. The sections contains a description of the rules applying.
- **Validation of Derived Field Definitions** page 132

If the format of derived fields is changed manually, Predict performs validation checks. These checks are described in this section.
- **Defining Derived Fields of Special Types** page 125

The rules applying when defining specific types of derived fields are described:

 - Superfields/Descriptors, page 125
 - Subfields/Descriptors, page 127
 - Phonetic Descriptors, page 133
 - Hyperdescriptors, page 134
 - Key or Index Fields in SQL Files, page 135
 - VSAM Primary Superdescriptors or Alternate Indices, page 137

General Rules for Defining Derived Fields

If a derived field is modified, a table containing the source fields appears when ENTER is pressed in the *Modify Field* screen. The size and format of this table varies with the type of derived field.

The table in the screen below is for a superfield. Editor functions and a selection mechanism help when defining derived fields.

12:04:29

***** P R E D I C T 4.1.1 *****

1999-02-25

- Modify Field -

Field ID ARH_SP

Modified 1999-01-07 at 16:14

File ID ARH-D1

Ty	L	Field name	F	Length	Occ	D	U	DB	N	NAT-1
SP	1	ARH_SP				D	U	AB		

Index Name ARH-ARH_SP

Source field name *

A/D

1

ARH1

A

2

ARH3

D

3

ARH2

A

4

ARH4

A

5

ARH5

A

6

ARH6

A

7

ARH7

A

8

ARH8

A

9

ARH9

A

10

ARH10

A

EDIT:

Owner: N

Desc: N

Veri: N

MORE

Attr.: N

Scroll to: __

General Attributes of Definitions of Derived Fields

The following attributes are contained in most definitions of derived fields. Attributes specific to certain types of derived fields are described in the respective sections.

Editor Functions

Deleting Source fields

Source fields can be deleted from the definition by overwriting their name with blanks. The remaining lines will be reorganized automatically.

Moving Source fields

Source fields can be moved with the *.m* command. Enter *.m* at the beginning of the line to be moved, position the cursor in the line where the moved line is to appear, and press ENTER. The table of source fields is automatically reorganized.

Scrolling

If a definition of a derived field contains more source fields than can be displayed in one screen, the source field to be displayed on top of the list can be specified in the field *Scroll to*. See page 123.

Selection Mechanism

New source fields can be added to the definition of a field by selecting them from a list of all fields contained in the file. This list is displayed in a *Source field* window if a name with asterisk notation (*) is entered as selection criterion in the list of the current definitions (as shown in the screen below).

```

12:09:01          ***** P R E D I C T  4.1.1  *****          1999-02-25
                                - Modify Field -
Field ID ..... ARH_SP          +Top-----Source field-----+
File ID ..... ARH-D1          ! _ ARH1                      !
                                ! _ ARH2                      !
                                ! _ ARH3                      !
Ty L Field name          F ! _ ARH4                      !
-- - ----- -          ! _ ARH5                      !
SP 1 ARH_SP              ! _ ARH6                      !
                                ! _ ARH7                      !
                                ! _ ARH8                      !
      Index Name .... ARH-ARH_SP ! _ ARH9                      !
      Source field name *          ! _ ARH10                     !
1      ARH1              ! _ ARH11                     !
2      ARH3              ! _ ARH12                     !
3      ARH2              ! _ ARH13                     !
4      ARH4              ! _ ARH14                     !
5      ARH5              ! _ ARH15                     !
6      *RH6              ! _ ARH16                     !
7      ARH7              ! _ ARH17                     !
8      ARH8              ! _ ARH18                     !
9      ARH9              ! Command ==> +_____          !
10     ARH10             !
EDIT:  Owner: N   Desc: N   Veri: N   +More-----+

```

A source field is selected by marking it in the left column or by positioning the cursor in the respective line and pressing ENTER. One field can be selected at a time.

Defining Derived Fields of Special Types

- This section is organized as follows:
- Superfields/Descriptors for Files of Type A, C and Z, page 125
 - Subfields/Descriptors for Files of Type A, C and Z, page 127
 - Phonetic Descriptors for Files of Type A, C and Z, page 133
 - Hyperdescriptors, page 134
 - Key or Index Fields in SQL Files (Superfields), page 135
 - VSAM Primary Superindex or Alternate Superindex, page 137

Superfields/Descriptors for Files of Type A, C and Z

The screen for the definition of superfields/descriptors for files of type *Adabas C*, *Conceptual* and *Standard* looks as follows.

09:13:06

***** P R E D I C T 4.1.1 *****

1999-02-13

- Modify Field -

Field ID SUPER-1

Added 1997-06-07 at 12:39

File ID PD-A2

Modified 1998-02-12 at 10:53

Ty	L	Field name	F	Length	Occ	D	U	DB	N	NAT-1
SP	1	SUPER-1	A	168.0					AJ	N

Source field name *

1

STD-EL1

A

30.0

1

10

2

STD-EL2

N

23.0

2

2

3

STD-EL2

N

23.0

2

3

4

LOGICAL

B

10.0

5

LOGICAL

B

10.0

6

MAIG

A

50.0

7

TIME

T

8

Start/End: Relative byte position in source field. Default
is first and last byte of source field

EDIT: *

Owner: N

Desc: Y

Veri: N

MORE

Attr.: N

Scroll to: __

Attributes

Source field name	Name of the fields used by derived fields.
F, Length	Format and length of the source field. These columns are read-only. Section Rules Applying to Format Changes , page 128, describes how the format of the derived field is determined by Predict.
Start	The relative byte position where the part of the source field to be used by the derived field starts (not applicable to phonetic descriptors). See also Specifying the Start and End Position below.
End	The relative byte position where the part of the source field to be used by the derived field ends (not applicable to phonetic descriptors and VSAM Primary Superdescriptors or Alternate Indices). See also Specifying the Start and End Position below.

Specifying the Start and End Position

The start and end values given in the definition of a derived field are always byte positions within the source fields (beginning with 1 and counting from left to right for alphanumeric fields and binary fields and from right to left for numeric fields).

The full length is used if no start and end values are specified.

In Adabas it is possible to address byte positions outside of the length of field. If this feature is used and a start byte outside of the source field specified, an end byte must be specified.

Note:

Special rules apply when specifying the length of subfields/descriptors. See page 127.

The following rules apply:

- Superfields/descriptor definitions can be based on up to twenty source fields.
- Only formats *A*, *B* and *N* are possible for superfields/descriptors.
- Adabas recognizes only format *A* and *B* for this type of field.
- Format *N* can be useful for Natural, but is not recommended because an alphanumeric or binary value cannot be converted to a numeric field.

Note:

See also **General Rules of Defining Derived Fields**, page 123.

Subfields/Descriptors for Files of Type A, C and Z

Subfields/Descriptors for files of type *Adabas C*, *Conceptual* and *Standard* are defined in the screen below.

09:12:28

***** P R E D I C T 4.1.1 *****

1999-06-30

- Add a Field -

Field ID PHON-4

Added 1998-06-30 at 09:11

File ID HEB-FI

Modified

Ty	L	Field name	F	Length	Occ	D	U	D	B	N	NAT-1
SB	1	PHON-4									SS N

Source field name *

F Length Start End

1 HEB-TEST A 1.0

Start/End: relative byte position in source field. Default

is first and last byte of source field.

EDIT:

Owner: N

Desc: N

Veri: N

MORE

Attr.: N

Scroll to: __

With subfields/descriptors, only one source field can be entered in the window.

Specifying the Length of Subfields

If the source field of a subfield/descriptor has format *P* and the start byte is greater than 1, the length of the subfield/descriptor is *normal length+1*, because the sign of the source field is always included in the subfield/descriptor field (see *Adabas Utilities Manual*).

Example:

Given that

- the source field has format P and length 5,
- the subfield/descriptor definition is source field from 2 to 3,
- the length of the subfield is 3 bytes (2 bytes + 1 byte for sign),
- the 3 bytes packed are 5 digits,

then the subfield/descriptor has format P and length 5.

Note:

See also **General Rules for Defining Derived Fields**, page 123.

Rules Applying to Format Changes

Note:

To understand the following, some knowledge of the hierarchical data structures of Predict and the process of rippling is required. See **Rippling**, page 264, in Chapter **File** for more information.

Determining the Format of Sub/Superfields/Descriptors

The format of sub/superfields/descriptors in files of type *Adabas C*, *Conceptual* and *Standard* (codes *A*, *C*, *Z*) is generated automatically by Predict.

A format of a derived field that has been determined by Predict can, however, be overwritten manually.

The following sections describe the rules applying.

Subfield/Descriptor

Subfield/descriptors always have the same format than the source fields they are derived from. If the format of a source field is changed, the format of the subfield/descriptor is changed accordingly.

Superfield/Descriptor without Format

If a superfield/descriptor is defined without a format, Predict assigns the format as follows:

- Format=*A*
if at least one source field of the SP field is defined with format *A*, or
if one of the source fields specified in the definition does not yet exist in the file.
- Format=*B*
if no source field is defined with format *A*.

Superfield/Descriptor with Format

If the format of source fields has been changed, Predict checks if the new and the old format of the source field are compatible. If they are compatible, the change does not have any impact on the format of the superfield/descriptor.

The formats *NS*, *US*, *N* and *U* and the formats *P* and *PS* are compatible. So, if the format is changed from *N* to *US*, for example, the format of the superfield/descriptor will not change.

If the new and the old format of the source field are **not** compatible, a window appears in which a format change proposed by Predict can be confirmed or a new format can explicitly be assigned to the superfield/descriptor (see screen below).

```
13:51:16          ***** P R E D I C T  4.1.1  *****          1999-02-13
                        - Modify field -
Field ID ..... FELD5                      Added 1998-02-13 at 13:15
File ID ..... JCA-STAL                    Modified 1998-02-13 at 13:15

Ty L Field-name                               F Length Occ D U DB N NAT-1
-----
SP  FELD5                                     B    25.0

      +-----+
      ! Superdescriptor definition changed      !
      ! to the correct format.                  !
      !                                         !
      ! old format .. B    new format .. A      !
      !                                         !
      ! Hit 'ENTER' to continue or change format. !
      +-----+
```

Changing the Format of Superfield/Descriptors Manually

The format of a superfield/descriptor can be changed manually (with the *Modify Field* function). If a source field of the superfield/descriptor is then changed again, Predict checks if the change affects the format of the superfield/descriptor.

The Impact of Changes to Standard Fields (Rippling)

Changes to sub/superfield/descriptors and fields used in sub/superfield/descriptors (source fields) are rippled as described in the sections below.

Changes to Sub/Superfield/Descriptors

It is not recommended to define sub/superfield/descriptors in standard files and to use these in real files. It is however possible to do it. The following rule then applies:

Note:
Changes to the format and length and changes to the definition of derived fields in standard files are not rippled from standard files to real files and userviews.
This is because the definition of derived fields is not coupled, and rippling format and length alone could lead to inconsistent data definitions in real files and userviews.

Changes to Source fields

Changes to the format of a standard field are rippled as normal to all fields in a file connected to this standard field.

If a field in an Adabas C file is used in the definition of a sub/superfield/descriptor, the format of the sub/superfield/descriptor is also changed if one of the following conditions are met:

- the resulting format is *A*, or
- the resulting format is *B* and the old format was *A*.

Note:

In the case of superdescriptors, if the format in the Adabas C file is set (manually) to *N* and the correct format were *B*, no change is made (unless the field length is greater than 29).

How the Rippling of Changes to Source fields is Indicated

If changes to standard fields are rippled to derived fields in real files and userviews, two screens are displayed indicating this process of rippling.

In the first screen the **changes of source fields** are indicated.

```
13:51:35          ***** P R E D I C T  4.1.1  *****          1999-02-13
                        - Modify file -                               Page:  1

                        list of field updates
                        -----

FELD2                *** upd ***
FELD2                JCA-ADA1          *** upd ***
FELD5                JCA-ADA1          *** upd ***
FELD5                *** upd ***
```

In the second screen the **changes of the format and/or length** of derived fields are indicated.

```
SUB/SUPER/PHON- fields, -descriptors length are changed
-----

Ty Field name          File name
-----
SP FELD5              JCA-STAl          updated
```

Validation of Derived Field Definitions

Predict performs the following validations for derived fields:

- A superfield/descriptor can have only one source field which is a multiple-value field.
- Source fields with format *D*, *T*, or *L* must not have a start or end character.

The following rule applies for all file types except *Conceptual* and *Standard*:

All source fields must exist in the file. This check is performed when a CHECK or CAT command is entered in the field list editor or when the *Add/Copy/Modify field* function is executed from the *Field Maintenance* menu.

These validations can be executed differently:

- If a source field is changed with the list editor (function *Link children* in the *Modify File* menu with *Related type* set to *EL*), the validation can be executed explicitly with the CHECK command.
The CAT command will also perform the validation.
- If a source field is changed with the function *Modify Field*, the validations are performed directly.

Phonetic Descriptors for Files of Type A, C and Z

The screen for defining phonetic descriptors for files of type *Adabas C*, *Conceptual* and *Standard* is identical to that for subfields/descriptors on page 127.

With phonetic descriptors, only one source field can be entered in the window.

The *Start* and *End* attributes do not apply to phonetic descriptors: Adabas always uses the first 20 bytes of this field to build a phonetic descriptor.

Further information on sub/superfields/descriptors and phonetic descriptors can be found in the chapter **ADACMP (COMPRESS-DECOMPRESS)** in the *Adabas Utilities Manual*.

Note:

See also **General Rules for Defining Derived Fields**, page 123.

Hyperdescriptors for Files of Type A, C and Z

The screen for defining hyperdescriptors looks as follows:

14:00:05

***** P R E D I C T 4.1.1 *****

1999-07-22

- Add a Field -

Field ID FIELD3

Added 1998-07-22 at 13:59

File ID DEMO

Modified

Ty	L	Field name	F	Length	Occ	D	U	DB	N	NAT-1
HY	1	FIELD3	A	20.0		D		XZ	N	

User exit nr ...

Source field name *

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

EDIT:

Owner: N

Desc: N

Veri: N

MORE

Attr.: N

Attributes

User exit nr

A number between 1 and 31 identifying the user exit that defines the hyperdescriptor. See Chapter **User Exits** in the *Adabas DBA Reference Manual*.

Key or Index Fields in SQL Files (Superfields)

The screen below is used for defining Keys or Indexes in fields of the following file types:

- A Adabas C file (with parameter *Adabas C SQL usage* set to *Y*)
- BT, BV Adabas D table / view
- D, E DB2 table / view
- JT, JV INGRES table / view
- OT, OV ORACLE table / view
- X General SQL
- XT, XV INFORMIX table / view
- YT, YV SYBASE table / view

The following rules apply:

- If the field type is *blank* (normal field), the key or index is based on one field
- if the field type is *SP* (superfield), the key or index includes more than one field.

15:10:07

***** P R E D I C T 4.1.1 *****

1999-06-21

- Modify Field -

Field ID ARHSP

Modified 1998-06-07 at 16:14

File ID ARH-DB2

by ARH

Ty	L	Field name	F	Length	Occ	D	U	DB	N	NAT-1
SP	1	ARHSP				D		AM		

Index Name ARH-ARHSP

Source field name *

A/D

1 ARH7 A

2 ARH8 D

3

4

5

6

7

8

9

10

EDIT:

Owner: N

Desc: N

Veri: N

MORE * Attr.: Y

Scroll to: __

Attributes

Index name	The name of the key or index. Must be entered in qualified form: creator/schema name followed by key or index name, separated by a hyphen. The creator/schema and key or index name are subject to SQL naming conventions (see page 135). Creator name and field name are concatenated and proposed as index name.
Source field name	The name of a column (source field) from which the key or index is derived. If the key or index is based on one field (field type <i>blank</i>), the name of that field is displayed and cannot be changed. If the key or index includes more than one field (Field type <i>SP</i>), up to 20 column names can be entered. Each must name a column of the table.
	<i>Note:</i> For fields in files of type X (General SQL), you can enter up to 16 column names.
	Enter a value in the <i>Scroll</i> field to define source fields greater than 10.
A/D	<p>A Puts key or index entries in ascending order by source fields (column). Default.</p> <p>D Puts key or index entries in descending order by source fields (column).</p>

VSAM Primary Superindex or Alternate Superindex

VSAM superdescriptors (Field type *SP*) in a file of type *V* (physical VSAM) and *L* (logical VSAM) are defined in the following screen.

14:45:57***** P R E D I C T 4.1.1 *****1999-07-22

- Modify Field -

Field ID KEYAdded 1997-07-05 at 00:00

File ID PD-V1Modified 1998-06-03 at 16:42

Ty	L	Field name	F	Length	Occ	D	U	DB	N	NAT-1
SP	1	KEY		30.0		A		AE		N

	Source field name *	F	Length	Start	End
1					

Start/End: relative byte position in source field. Default is first and last byte of source field.

EDIT: Owner: N Desc: N Veri: N MORE Attr.: N Scroll to: __

Attributes

Start The starting position (offset plus one) of the superdescriptor within the source field. An end position cannot be specified.

If a VSAM field on an alternate index (descriptor *A*) in a file of type *V* (physical VSAM), *L* (logical VSAM), or *C* (conceptual) is defined (the descriptor type is *A*), a second screen is displayed for defining additional attributes: *upgrade flag*, *sort flag*, *null flag* and *DD name* (see below).

Additional Attributes for VSAM Alternate Fields

14:44:37

***** P R E D I C T 4.1.1 *****

1999-07-22

- Modify Field -

Field ID KEY

Added 1997-07-05 at 00:00

File ID PD-V1

Modified 1998-06-03 at 16:42

Keys ..

Zoom: N

Ty	L	Field name	F	Length	Occ	D	U	DB	N	NAT-1
*- - - - -			*- - - - -			*	*	- -	*	- - - -
1		KEY	A	30.0		A		AE	N	

NATURAL attributes

Header1

Header2

Header3

Edit mask ..

Comments Zoom: N

+ - VSAM descriptor attributes - +

! Upgrade flag ... Y (Y,N) !

! Sort flag N (Y,N) !

! Null flag N (Y,N) !

! DD name DDNAME !

+ - - - - - +

EDIT: Owner: N Desc: N Veri: N

MORE Attr.: N

Additional Descriptor Attributes

Upgrade Flag	Y	Alternate index is updated by Natural.
	N	Alternate index is updated by VSAM.
Sort Flag	Y	If the upgrade flag is also Y, the alternate index is read in ascending order. Otherwise, the alternate index is read in the order in which the values were entered during field update.
Null Flag	Y	Records with a null value in this index field are suppressed.
DD Name		The DD name associated with this alternate index file. In CICS, the FCT name of the VSAM file.

Defining More Attributes of Fields

If *MORE Attr.* is set to *Y*, a window is displayed containing additional attributes for selection.

- 3GL specification, page 140
- Field name synonyms, page 144
- Old mode synonym, page 145
- Condition name & value, page 142
- Adabas security & Edit mask, page 146
- Field procedure, page 147
- Derived field expression, page 148
- Index Definition (DB2), page 149
- Default name, page 153
- Constraint name, page 154

The following rules apply:

- Only those types of additional attributes appear in the window that apply to the type of field. For example: the option *Adabas security & Edit mask* is not contained in the list when a DB2 index field is processed.
- More than one choice can be made at a time. The respective input maps are then displayed one after the other.

The additional attributes are described in the following sections.

3GL Specification

```
14:06:25          ***** P R E D I C T 4.1.1 *****          1999-03-06
                        - Modify Field -
Field ID ..... JCA-EL1                      Added 1999-04-20 at 14:55
File ID ..... JCA-FIL                      by JCA

Ty L Field ID                                     F Length   Occ   D U DB N
*- - - - - * - - - - - * - - - - - * - - - - -
  1 JCA-EL1                                     A   2.0           AA N

Specifications for 3GL
Gr.structur ..... (n)
Justify ..... (R)
Synchronized ..... (S)
Initialize with ...*
  Init value .....
Indexed by .....
Depending on .....
```

Attributes

- Gr.structur

The field attribute *Gr.structur* is used to change the record layout generated from a PE/PC field.
If *Gr.structur* is set to *N*, all fields within a PE group are treated as multiple value fields. Setting *Gr.structur* to *N* prevents the format buffer for Adabas from becoming very large.
Gr.structur = *N* can only be specified for real fields in the deepest PE group (highest level number). For example: if there are 3 PE groups in the file on level 1, 4 and 6, only the PE groups on level 6 can be marked with *Gr.structur* = *N*.

If *Gr.structur* is set to *blank*, PE/PC groups are to be generated as groups which occur *n* times as a whole.
- Justify

R When COBOL copy code is generated, the statement JUSTIFIED RIGHT is added for this field. Any data written to this field is then right-justified.
L Data will be left-justified. Default.

Synchronized	<p>Applicable to fields of type <i>I</i>, <i>F</i> or <i>B</i> and length 1, 2, 4 or 8.</p> <p>S when Assembler, COBOL or PL/I copy/include code or a record layout is generated, this field can be aligned on a half-word, word, or double-word boundary (speeding up arithmetic operations). This affects format buffer generation and the offsets of the fields in the record buffer. Slack-bytes are inserted into the record buffer by the assembler or compiler but they are built into any format buffer by Predict using space characters <i>X</i>.</p>														
Initialize with	<p>Determines the initial value for generation. To be used instead of the standard value (zeros for a numeric field, blanks for an alphanumeric field).</p> <table><tr><td>S</td><td>blank</td></tr><tr><td>L</td><td>low value</td></tr><tr><td>H</td><td>high value</td></tr><tr><td>Z</td><td>zero</td></tr><tr><td>Q</td><td>quote</td></tr><tr><td>F</td><td>Fill with string specified in the parameter <i>Init. value</i> (mandatory). For example: if <i>X</i> is specified an the field length is 4, <i>XXXX</i> will be used for initialization.</td></tr><tr><td><i>blank</i></td><td>Field will be initialized with the string specified in the field <i>Init. value</i>. If no <i>Init. value</i> is specified, no initialization is performed.</td></tr></table>	S	blank	L	low value	H	high value	Z	zero	Q	quote	F	Fill with string specified in the parameter <i>Init. value</i> (mandatory). For example: if <i>X</i> is specified an the field length is 4, <i>XXXX</i> will be used for initialization.	<i>blank</i>	Field will be initialized with the string specified in the field <i>Init. value</i> . If no <i>Init. value</i> is specified, no initialization is performed.
S	blank														
L	low value														
H	high value														
Z	zero														
Q	quote														
F	Fill with string specified in the parameter <i>Init. value</i> (mandatory). For example: if <i>X</i> is specified an the field length is 4, <i>XXXX</i> will be used for initialization.														
<i>blank</i>	Field will be initialized with the string specified in the field <i>Init. value</i> . If no <i>Init. value</i> is specified, no initialization is performed.														
Init. value	<p>If <i>Initialize with</i> is either <i>F</i> or <i>blank</i> a value used for initialization of a field must/can be specified.</p> <p>Length and format of the <i>Init value</i> must be valid for the field. For binary fields hexadecimal constants such as <i>FB0A</i> are valid.</p> <p>See also Condition Name & Value below.</p>														
Indexed by	<p>String that is used when generating the COBOL INDEXED BY clause (only valid for fields of type MU/MC or PE/PC).</p>														
Depending on	<p>String used when generating the COBOL DEPENDING ON clause (only valid for fields of type MU/MC or PE/PC).</p>														

Condition Name & Value

14:04:26

***** P R E D I C T 4.1.1 *****

1999-09-13

- Modify Field -

Field ID VE-FIELD

Modified 1999-08-27 at 11:21

File ID HEB-A

by HEB

Ty	L	Field ID	F	Length	Occ	D	U	D B	N
*-	-	-----	*	-----	-----	*	*	--	*
1		VE-FIELD	A	3.0				AA	N

Condition name

FC * Condition value

EDIT:

Owner: N

Desc: N * Veri: Y

Scroll to: 1

Attributes

Condition name

A value to be used when generating either equate data in Assembler copy code or a level 88 entry in COBOL copy code.
Up to 10,000 condition names can be entered. Each name needs at least one corresponding condition value. Using condition names can make logical conditions and assignments easier to handle.

FC	<p>Figurative constant. Valid values:</p> <table><tr><td>S</td><td>blank</td></tr><tr><td>L</td><td>low value</td></tr><tr><td>H</td><td>high value</td></tr><tr><td>Z</td><td>zero</td></tr><tr><td>Q</td><td>quote</td></tr><tr><td>F</td><td>Fill with string specified in the parameter <i>Condition. value</i>. For example: if <i>X</i> is specified an the field length is 4, <i>XXXX</i> is used as condition value.</td></tr><tr><td><i>blank</i></td><td>The string specified in the field <i>Condition. value</i> is used.</td></tr></table>	S	blank	L	low value	H	high value	Z	zero	Q	quote	F	Fill with string specified in the parameter <i>Condition. value</i> . For example: if <i>X</i> is specified an the field length is 4, <i>XXXX</i> is used as condition value.	<i>blank</i>	The string specified in the field <i>Condition. value</i> is used.
S	blank														
L	low value														
H	high value														
Z	zero														
Q	quote														
F	Fill with string specified in the parameter <i>Condition. value</i> . For example: if <i>X</i> is specified an the field length is 4, <i>XXXX</i> is used as condition value.														
<i>blank</i>	The string specified in the field <i>Condition. value</i> is used.														
Condition value	<p>The length and format of this value must be valid for this field. This value must have a corresponding condition name. Up to ten condition values can be entered. If several values correspond to the same name, put the name before the first value and leave the name field blank before later values. THRU in the name field indicates a range of values ending with the value on that line and beginning with the value on the previous line.</p>														

Field Name Synonyms

14:48:12

***** P R E D I C T 4.1.1 *****

1999-07-22

- Modify Field -

Field ID PD-A2

Added 1997-08-24 at 00:00

File ID PD-A-TEST3

Modified 1998-05-07 at 18:08

Ty	L	Field name	F	Length	Occ	D	U	DB	N	NAT-1
*	-	-----	*	-----	-----	*	*	---	*	-----
	2	PD-A2	A	2.0					AF	

Field name synonyms

User defined

NATURAL

COBOL

PL/I

BAL/ASSEMBLER

FORTRAN

PASCAL

Language ADA

Language C

EDIT: * Owner: N Desc: N Veri: N

Attributes

Field name synonyms

Synonyms to be assigned to the field when definitions in the following programing languages are generated: Natural, COBOL, PL/I, BAL (Assembler), FORTRAN, PASCAL, ADA or C.

Old Mode Synonyms

- This option is only provided for compatibility with old versions of Predict.
- Whether this option is displayed depends on the parameter *Old mode synonyms* of the screen *Defaults ->General defaults -> Synonyms* :
- N Default setting. Compatibility with old versions is not required.
 - Y Up to 90 synonyms can be defined as Natural synonyms. These create additional entries in the DDM which have the same attributes as the original object but different names.

15:10:46

***** P R E D I C T 4.1.1 *****

1999-10-26

- Modify Field -

Field ID PDS

Modified 1999-10-14 at 16:30

File ID PD-D1

by PD

Field synonym

Field synonym

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

EDIT:

Owner: N

Desc: N * Veri: N

More:

Synonyms: N

Adabas Security & Edit mask

14:48:35

***** P R E D I C T 4.1.1 *****

1999-07-22

- Modify Field -

Field ID PD-A2

Added 1997-08-24 at 00:00

File ID PD-A-TEST3

Modified 1998-05-07 at 18:08

Ty	L	Field name	F	Length	Occ	D	U	DB	N	NAT-1
*	-	-----	*	-----	-----	*	*	---	*	-----
2		PD-A2	A	2.0						AF

ADABAS attributes

Edit mask

Security access level .. (0-15)

Security update level .. (0-15)

EDIT: * Owner: N Desc: N Veri: N

Attributes

Edit mask	<p>The Adabas edit mask to be used for the field. Determines how numeric fields are to be edited. Valid values: E1...E15.</p> <p><i>Edit mask</i> is supported for compatibility reasons and for documentation purposes only.</p> <p>See section Format Buffer Syntax in the <i>Adabas 5 Command Reference Manual</i> for more information.</p>
Security access level	The Adabas access security level of the field.
Security update level	The Adabas update security level of the field.

Field Procedure

14:50:45

***** P R E D I C T 4.1.1 *****

1999-07-22

- Modify Field -

Field ID A_L10

Added 1997-07-07 at 00:00

File ID PD-DB2

Modified 1998-05-17 at 12:40

Ty	L	Field name	F	Length	Occ	D	U	DB	N	NAT-1
*	-	-----	*	-----	-----	*	*	-	*	-----
1		A_L10	AL	100.0		D		AC	N	

+-----+

! Procedure name

! Procedure parameter

!

!

!

!

!

!

!

!

+-----+

EDIT: * Owner: N Desc: N Veri: N

Attributes

Procedure name	Name of a field procedure (DB2 parameter FIELDPROC). See the DB2 documentation for more details.
Procedure parameter	Parameters passed to the field procedure. See the DB2 documentation for more details.

Derived Field Expression

```
>                                     > + EL: A5                                L: 1    S: 1
All  ....+....1... Expression for derived field...+....5....+....6....+....7..

*
* Field expression of a derived field
*
USER-TABLE1-SALARY * 12          /* SALARY FOR 12 MONTHS
+ CORRELATOR2-BONUS
```

Applicable only to fields of type *DV* in files of the following types:

B	Adabas C SQL views	X	General SQL
E	DB2 view	XV	INFORMIX view
JV	INGRES view	YV	SYBASE view
OV	ORACLE view		

The expression used to derive the field is to be edited using one of the following depending on the your settings in the *Profile > Handling* screen:

- the Natural-based Subquery Editor, or
- the SAG Editor

The editor can also be called with

- function *Edit Field expression* (Code *Y*) in the *Field Maintenance Menu*, or
- command EDIT ELEMENT EXPRESSION <file-id> <field-id> or

See Chapter **Editors in Predict** in the *Predict Reference Manual* for more information.

The subquery of the file that contains the current field can specify a correlation name for any file whose fields it uses. The name of each field referenced in the expression must be qualified (preceded) by the correlation name of the file from which the field is taken, if a correlation name has been specified for that file, or the ID of the file from which the field is taken, if no correlation name has been specified for it. The expression can include both comment lines (with /*, * or ** in the first two columns) and line comments (preceded by /*).

Example: A field which contains the annual salary:

```
*
* Field expression of a derived field
*
USER-TABLE1-SALARY * 12          /* SALARY FOR 12 MONTHS
+ CORRELATOR2-BONUS
```

Index Definition (DB2)

Index fields (descriptor type *D*, *F* or *P*) in a file of type *D* (DB2 table), are defined in the screen below.

Screen 1

15:38:13***** P R E D I C T 4.1.1 *****1999-06-22

- Modify Field -

Field ID PART_INDEXModified 1998-06-22 at 15:38

File ID SMR-PARTITIONEDby JCA

Definition of Index

Index name SMR-PARTITIONED_INDEXNumber of partitions: 3

Cluster index Y (Y/N)Piece Size ...* 256

subpages 4

close option Y

bufferpool* BP0

password required .. Y (Y/N)

index type

Default values of using- and free-block

VSAM catalog name ..

Storagespace*

Primary alloc

Secondary alloc.....

Erase opt (Y/N)

Free pages 0

Percentage free

GBPCACHE

EDIT: Owner: N * Desc: N Veri: N

Attributes

Definition of Index	
Index name	The name of the DB2 index. See page 136. A read-only field.
Number of partitions	Number of partitions (if greater than zero). A read-only field.
Cluster index	Y The records (rows) in the DB2 table are stored in the sequence of this index. Valid for max. one index per table. A table contained in a partitioned tablespace must have one index marked as a clustered index.

Subpages	The number of subpages for each physical page. The subpage is the unit of index locking in DB2. Valid values: 1, 2, 4, 8, 16.
Bufferpool	The buffer pool associated with the index.
Close option	Y The data sets supporting this index are closed when nobody uses the index.
Password required	Y A password should be specified for the CREATE INDEX statement. If the option <i>Passwords for indexes</i> in the <i>Generate table</i> function is set to <i>S</i> (special), a password can only be entered for indexes which have <i>Password required</i> set to <i>Y</i> .
Index type	Leave this field blank for the default value of the DB2 installation or enter: 1 Index type 1 2 Index type 2.
Piece size	The maximum piece size for a non-partitioned index. Valid values: 0, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, 262144, 524288, 1048576, 2097152, 4194304.
Default Values of Using- and Free-Block	
	The parameter <i>VSAM catalog name</i> must be specified if data sets are already defined for the index. Attributes marked * apply if data sets for the index have yet to be defined by DB2. The parameters <i>Free pages</i> and <i>Percentage free</i> apply in both cases.
VSAM catalog name	Name of the VSAM catalog for the index.
Storage group	Storage group where DB2 defines the data sets for the index (optional). If no storage group is specified, DB2 uses the default storage group.
Primary alloc	Minimum primary space allocation (in Kbyte) for DB2-defined index data sets. A value specified in this field is stored only if the attribute <i>storage group</i> has been specified.

Secondary	<p>Minimum secondary space allocation (in Kbyte) for DB2-defined index data sets.</p> <p>A value specified in this field is stored only if the attribute <i>storage group</i> has been specified.</p>
Erase opt	<p>Y The DB2-defined data sets are to be erased (filled with nulls) when the index is dropped.</p> <p>A value specified in this field is stored only if the attribute <i>Storage group</i> has been specified.</p>
Free pages	<p>A number from 0 to 255 which indicates that one page is to be left free each time this number of pages is used when the load operation creates index entries or when the index is reorganized.</p> <p>Zero indicates that no pages are to be left free.</p>
Percentage free	<p>A number from 0 to 99: the percentage of each page to be left as free space when index entries are created by a load operation or when the index is reorganized.</p>
GBPCACHE	<p>Only relevant in a data sharing environment. Specifies what pages of the table space or partition are written to the group buffer pool.</p> <p>Leave this field blank or enter:</p> <p>C Changed. Only pages that have been changed are written to the group buffer pool.</p> <p>A All pages are written.</p>

Screen 2

For a partitioned index (a cluster index for a table in a partitioned table space), the following screen is displayed for every two partitions. Each partition can then be defined in accordance with the *Default values of using- and free-block* (see description above).

```

14:41:51          ***** P R E D I C T 4.1.1 ***** 1999-09-19
              - Modify Field -

Field ID ..... EINS                               Modified 1999-07-28 at 09:21
File ID ..... SMR-D_MIT_INDEX                      by SMR

----- Definition of partitioned Index -----
Partition 1
  Value ..... 'a','b','c'

  VSAM catalog name .....
  Storagespace .....* SYSDEFLT
  Primary allocation .... 12                      GBPCACHE .....*
  Secondary allocation .. 12                      Free pages .....
  Erase option ..... N (Y/N)                     Percentage free .. 10

Partition 2
  Value ..... 'b','a','c'

  VSAM catalog name .....
  Storagespace .....* SYSDEFLT
  Primary allocation .... 12                      GBPCACHE .....*
  Secondary allocation .. 12                      Free pages .....
  Erase option ..... N (Y/N)                     Percentage free .. 10

EDIT:  Owner: N   Desc: N * Veri: Y                MORE * Partition: Y

```

Attributes

Value	<p>The highest value of the index key in this partition.</p> <p>At least one constant must be used and as many constants as there are columns in the key can be specified.</p> <p>The concatenation of all the constants is the highest value of the key in this partition of the index.</p>
-------	--

Note: No checking is performed here.

All other attributes are described above.

Default name

- This additional attribute is only applicable for fields in
- SYBASE tables with *Null value option* set to *R* and *Null default option* set to *Y*.
 - Adabas D tables, DB2 tables, Informix and Oracle tables with *Null value option* set to *R* or *U* and *Null default option* set to *Y*.

```
11:09:33          ***** P R E D I C T 4.1.1 *****          1999-10-04
                        - Modify Field -
Field ID ..... TESTFIELD          Modified 1998-10-04 at 11:09
File ID ..... JCA-YT              by JCA

Ty L Field ID          F Cs Length   Occ   D U DB N Df NAT-l
*- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  1 TESTFIELD          A  B      10.0          AA R Y

Default name .....
Default expression .....<

>
```

Attribute

Default name	The default specified here is used in the CREATE TABLE statement. SYBASE naming conventions apply. See page 201.
<i>Note:</i>	For SYBASE, a <i>default</i> is an object in its own right. For other SQL systems, a default value is specified in the CREATE TABLE statement (not null with default <i>default_expression</i>). For Informix no default name is allowed.
Default expression	An SQL expression can be specified between the angled brackets. This expression determines the default value, for example a constant or function. If specified, this value is always used by the function <i>Generate CREATE statement</i> .

Constraint name

Depending on the field definition, up to four constraint names can be specified.

11:35:45

***** P R E D I C T 4.1.1 *****

1999-10-04

- Modify Field -

Field ID TESTFIELD

Modified 1998-06-21 at 09:34

File ID JCA-YT

by JCA

Ty	L	Field ID	F	Cs	Length	Occ	D	U	DB	N	Df	NAT-1
*-	-	-----	*	-	-----	-----	*	*	-	-	*	-
1		TESTFIELD	A	B	10.0		P	U	AA	R	Y	

Attributes

Constraint name

Check constraint

Primary key

Unique

Null/Not null

Attributes

Check constraint	Constraint name in the respective SQL system for the fact that a linked verification of status <i>S</i> exists.
Primary key	Constraint name for the fact that the field is a primary key.
Unique	Constraint name for the fact that a unique constraint exists (indicated with <i>U</i> in column <i>Unique option</i> of the field object in Predict).
Null/Not null	Constraint name for the fact that the <i>Null</i> or <i>Not null default option</i> is set to <i>Y</i> .

Field Maintenance

The *Field Maintenance* screen is shown on page 93.

Note:

Predict does not perform complete consistency checks when executing field maintenance functions. It is therefore possible to spoil the integrity of field lists of files temporarily. Consistency checks are performed, however, when field lists are cataloged.

Predict maintenance functions applying particularly to objects of type field and aspects of standard maintenance functions that are specific to fields are described in the sections below. The following functions are described:

- *Add a Field*, page 156
- *Copy Field*, page 156
- *Move Field within a File*, page 157
- *Purge Field*, page 157
- *Redefine Field*, page 158
- *Browse through Fields of a File*, page 163
- *Link Verification*, page 163
- *Edit Field expression*, page 163

Note:

Standard maintenance functions are described in Chapter **Maintenance** in the *Predict Reference Manual*.

Add a Field (Code A)

The function *Add a Field* can be used to add a field

- to the end of the specified file (copy field ID=*blank*)
- to the top of the specified file (copy field ID=*)
- after a specified field (copy field ID=*Field_2*) in the specified file.

The position of a new field is determined with the parameters *Copy field ID* and *Copy file ID* as follows.

Parameter/Function	ADD to end	ADD to top	ADD to position
Field ID	<i>Field_1</i>	<i>Field_1</i>	<i>Field_1</i>
in file	<i>File_1</i>	<i>File_1</i>	<i>File_1</i>
Copy field ID	–	*	<i>Field_2</i>
Copy file ID	–	–	–

Command: ADD ELEMENT

Copy Field (Code C)

This function is useful for creating a new field entry. A field can be copied:

- to the end of the same file and renamed (copy field ID=*Field_2*)
- to another file (copy file ID=*File_2*) to a position after a specified field (copy field ID=*Field_2*)
- to the end of another file (copy file ID=*File_2*).

The position of a copied field is determined with the parameters *Copy field ID* and *Copy file ID* as follows.

Parameter/Function	COPY and rename	COPY to position	COPY to end
Field ID	<i>Field_1</i>	<i>Field_1</i>	<i>Field_1</i>
in file	<i>File_1</i>	<i>File_1</i>	<i>File_1</i>
Copy field ID	<i>Field_2</i>	<i>Field_2</i>	
Copy file ID	–	<i>File_2</i>	<i>File_2</i>

Command: COPY ELEMENT

Move Field within a File (Code *H*)

This function is used to change the order of fields in a file. A field can be moved:

- to the top of the file (copy field ID=*)
- to a position after a specified field (copy field ID=*Field_2*).

The position of a moved field is determined with the parameters *Copy field ID* and *Copy file ID* as follows.

Parameter/Function	MOVE to top	MOVE to position
Field ID	<i>Field_1</i>	<i>Field_1</i>
in file	<i>File_1</i>	<i>File_1</i>
Copy field ID	*	<i>Field_2</i>
Copy file ID	–	–

If the function *Move* is applied to group fields (type GR, PE, PC), all fields of the group will be moved.

Command: MOVE ELEMENT

Purge Field (Code *P*)

Predict objects of type *field* can be deleted with the *Purge* function (code *P*). The following rules apply.

- If the field to be deleted is a group, it is possible to delete all fields in the group.
- If the field is a standard field, connections to derived fields are deleted.
- If a field in a master file is deleted, all fields in userviews related to this field are deleted as well.

If you confirm the *Purge* function with DELETE,

- The field and its redefinition will be purged, and
- All file relations based on this field will be set to *Documented*.

Command: PURGE ELEMENT

Redefine Field (Code R)

The function *Redefine Field* (code *R*) invokes the list editor for defining a redefinition (see screen below). A redefinition must be cataloged. Predict reports errors in a redefinition.

Command: REDEFINE ELEMENT

Note:
No consistency check is performed for files of type *Conceptual* or *Standard*.

>									
> + Fi: JCA-FI1									
Ty	L	Field	ID		F	Length	Occ	L: 1	S: 1
*	-	-----			*	-	-	D U DB S	All
								* * - *	
1	JCA-EL1				A	2.0		AA N	

Field Types that can be Redefined

<i>blank</i>	Normal fields
DV	Derived fields (SQL)
GR	Group
HM	Hyperdescriptors as a multiple value fields
HP	Hyperdescriptors as a fields of a periodic group
HQ	Hyperdescriptors as a multiple value fields of a periodic group
HY	Hyperdescriptors
MC	Multiple value fields with automatic counter
MU	Multiple value fields
PC	Periodic group with automatic counter
PE	Periodic group
QN	SEQNO field
SB	Subfields/descriptors
SP	Superfields/descriptors

If a field of type MU, MC, PE or PC is redefined, the whole array (including all occurrences) is redefined. When redefining fields of these types, the occurrence number must be specified.

Position and Format of a Redefinition

A redefinition is started by defining a field of type RE having the same *level* and *Field ID* as the field to be redefined.

This field definition has to directly follow the redefined field/group.

Ty	L	Field ID	F	Length	Occ
*_	-	-----	*	-	-----
		1 FIELD-TEST	A	20	
RE		1 FIELD-TEST			
		2 TEST-REDEF1	A	5	
		2 FILLER	A	5	
		2 TEST-REDEF2	A	5	

Format and Type of Fields within the Redefinition

The following field types can be used within a redefinition: *blank*, MU, GR or PE. Redefinition within the redefinition is possible.

The number of occurrences must be specified for MU and PE fields. Format of occurrences are increased to N5. The occurrences can be specified in the *Edit Elements of a File* screen.

Properties of Fields within a Redefinition

Redefined fields have the following properties:

- The sum of the length of all fields in a redefinition must not be greater than the length of the field/group being redefined.
- The field level within a redefinition cannot be greater than 9.
- PE in PE is possible.
- The special field name FILLER is not tested for uniqueness and can be used to exclude parts of the original field from redefinition (as in previous versions of Predict).
- A field can be identified by a maximum of three indexes.
- Within redefinitions, *Gr.structur* must not be set to *N*.

Purge and Rename a Redefined Field

If a field that has been redefined is purged or renamed, all redefinitions of the field are purged or renamed as well.

Mapping of Natural Data Structures

Natural allows the definition of multiple arrays for one field. Data structures of this type cannot be defined in Predict and have to be circumscribed as shown in the two examples below.

Natural structure

```
1 GROUP      (1:2,1:4,1:3)
2 ELE      (A20)
```

Predict structure

```
PE 1 GROUP1      (2)
PE 2 GROUP1      (4)
PE 3 GROUP1      (3)
    4 ELE      A  20
```

Natural structure

```
1 ELE      (A20/1:2,1:4,1:3)
```

Predict structure

```
PE 1 GROUP1      (2)
PE 2 GROUP1      (4)
MU 3 ELE      A  20      (3)
```

Mapping of COBOL Data Structures

In Predict it is not possible to define new field attributes together with a redefinition (which is possible in COBOL).

An example of a COBOL structure using this feature and the Predict definition that is used to circumscribe the structure is shown below.

COBOL structure

```
01 FIELD-A    PIC X(A20).
01 FIELD-A-RED REDEFINES FIELD-A PIX X(1) OCCURS 10.
```

Predict structure

```
      1  FIELD-A          A 20.
RE 1  FIELD-A
MU 2  FIELD-A-RED      A 1  (10)
```

In COBOL it is not possible to redefine a PE or a MU field. An additional group field has to be inserted. Predict does this automatically when COBOL Copy Code is generated from a field in which a PE or a MU field is redefined.

An example of a Predict structure and the COBOL structure that circumscribes it is shown below.

Predict structure

```
MU 1  FIELD-MU          A 20  (10)
RE 1  FIELD-MU
      2  FIELD-MU-RED    A 200
```

COBOL structure

```
05 R-FIELD-MU
    10 FIELD-MU    PIC X(A20) OCCURS 10.
05 R-FIELD-MU-REGR REDEFINES R-FIELD-MU.
    10 FIELD-MU-RED PIX X(200).
```

Example

The following example defines the structure of a sequential file.

Typ	Lev	Field-name	F	Len	Occ	ADA
---	---	-----	--	---	---	---
*						
PE	1	PE-GROUP1			2	AA
PE	2	PE-GROUP2			3	AB
PE	3	PE-GROUP3			2	AC
	4	PE-EL1	A	6		AD
	4	PE-EL2	P	5.2		AE
RE	1	PE-GROUP1				
	2	PE-ELE-COMP	A	120		
*						
MU	1	MU-FIELD	A	250	5	AF
RE	1	MU-FIELD				
PE	2	PE-GR1			5	
MU	3	MU-FIELD1	A	5	4	
RE	3	MU-FIELD1				
	4	MU-FIELD1-1	A	15		
	4	MU-FIELD1-2	A	5		
	3	FLD01	A	10		
RE	3	FLD01				
	4	FLD01-1	A	5		
	4	FLD01-2	A	5		
RE	3	FLD01				
MU	4	FLD01-3	A	1	10	
	3	FILLER	A	2		
	3	FLD02	A	5		
	2	FLD03	A	20		
GR	1	GROUP				AG
	2	GR-ELE1	A	30		AH
	2	GR-ELE2	A	20		AI
RE	1	GROUP				
PE	2	GR-PE			50	
	3	GR-PE-EL	A	1		

Browse Through Fields of a File (Code B)

The *Browse through Fields of a File* function invokes the *Modify Field* function for each field in the field list of a file. If a field is specified in the parameter *Field ID*, the functions starts with this field.

The function is useful when applying general changes to all fields in a file.

Command: BROWSE ELEMENT

Link Verification (Code L)

The function invokes the Link Editor to edit the verification List of the field. Verifications can then be linked to or unlinked from fields.

Command: LINK ELEMENT VERIFICATION

Edit Field Expression (Code Y)

Depending on the editor preferences specified in the *Profile > Handling* screen, either the SAG Editor or the Natural-based Subquery Editor is called. See **Derived field Expression**, page 148.

Command: EDIT ELEMENT EXPRESSION

Field Retrieval

Field retrieval functions are called from the *Field Retrieval* menu, which is called with the command RETRIEVE ELEMENT or with Code *R* and object type code *EL* in a Predict *Main Menu*.

This section includes the following topics:

- Field-specific Retrieval Parameters, this page
- Sorting Files and Fields, page 166
- Field-specific Retrieval Functions, page 168
- Layout of Field Lists, page 170
- Output Options for Field Retrieval, page 172

Standard retrieval types are described in Chapter **Retrieval** in the *Predict Reference Manual*.

The retrieval types which retrieve information on parent objects (*with parents* / *with no parent*) do not apply to fields.

Field-Specific Retrieval Parameters

See also **Selection Criteria** and **Output Options** in Chapter **Retrieval** in the *Predict Reference Manual*.

Parameters for Selection

Field ID/Synonym	When retrieving information on fields, the identifiers of fields and language-specific synonyms can be used as selection criteria.
Synonym of language	Determines how <i>Field ID/Synonym</i> is used to select fields: <i>none</i> <i>Field ID/Synonym</i> applies to field IDs. <i>#</i> All: <i>Field ID/Synonym</i> applies to field IDs and to field name synonyms for all languages. <i>language</i> If any language is specified, <i>Field ID/Synonym</i> applies to field IDs and to field name synonyms of this language.

in file	ID of the file to which a field object belongs. <i>in File</i> has to be specified. See also Specifying Parameter Values in Chapter The User Interface in the Manual <i>Introduction to Predict</i> .
in files of type	Only fields contained in files of the specified type will be included in the selection. The value specified is stored in the global variables applying only to fields. See also Specifying Parameter Values in Chapter The User Interface in the Manual <i>Introduction to Predict</i> .

Output Options

3GL specification	Y The following 3GL-specific attributes of fields are displayed: <i>Gr.structur, Justify, Synchronized, Init. value, Indexed by, Depending on, Condition name</i> and <i>Condition value</i> .
Composed fields	Y The source fields of hyper/super/subfields are displayed when fields of these types are displayed.
Display length	The format in which the length of SQL fields is displayed. N Natural Format P Physical Format
DV–Field expression	Y Derived field expressions are displayed.
Linked verifications	Y Verifications linked to fields are displayed.
Natural options	Y Up to three headers displayed in Natural maps and the definition of the Natural edit mask are displayed.
Sorted by field	Used to determine how field and file lists are sorted: N Sort fields alphabetically by file ID. All fields are displayed in the order they are defined in the file. Y Sort fields alphabetically by field ID. Note that the sort order also depends on the selection criteria. See Sorting Fields and Files below for more information.
Synonyms	Synonyms of field names for specific languages are displayed. A language can be selected from a selection window.

Sorting Fields and Files

Field and file lists produced by retrieval operations can be sorted by field ID or by file ID.

Sorting by Field ID

If fields and files are sorted by field ID, fields that are used in different files are sorted alphabetically by file.

15:25:45	*****	P R E D I C T	4.1.1	*****	1999-01-17
		- List Field -			Page: 3
Cnt	Ty	L	Fieldname	F	Length D File ID
37	GR	1	A-BINARY-GROUP		TSH-C-FILE
38		1	A-CITY	A	20.0 * MISCELLANEOUS
39		1	A-CITY	A	20.0 D TNG-ADABAS-FILE1
40		1	A-CITY	A	20.0 TSH-C-FILE
41		1	A-DATE	D	TNG-ADABAS-FILE1
.		.		.	.
.		.		.	.

Field lists will be sorted by field if parameters are specified in one of the following combinations:

Sorted by Field	Field ID specified	File ID specified
Y		
Y	✓	
Y	✓	✓
N	✓	

Note:
If only a field id is specified as selection criteria, field and file lists are sorted by field, even if sorted by field is set to *N*.

Sorting by File ID

If fields are sorted by file, the fields appear in the order they are defined in the file.

15:29:12		***** P R E D I C T 4.1.1 *****				1999-01-17	
		- List Field -				Page: 1	
Cnt	Ty L	Fieldname	F	Length	D	File	ID
1	1	AA-FIELD	A	12.0	D *	A-ADDR-FILE	
2	1	AB-FIELD	A	1.0	D *	A-ADDR-FILE	
3	MU 1	AC-FIELD	A	20.0	D *	A-ADDR-FILE	
4	1	AD-FIELD	A	60.0	D *	A-ADDR-FILE	
5	1	AE-FIELD	A	60.0	*	A-ADDR-FILE	

Field lists will be sorted by files if parameters are specified in one of the following combinations:

Sorted by Field	Field ID specified	File ID specified
N		
N		✓
N	✓	✓
Y		✓

Note:
If only a file id is specified as a selection criterion, field and file lists are sorted by file even if sorted by field is set to Y.

Field-specific Retrieval Functions

- Fields and Related Views, page 168
- Non-Standard Fields, page 168
- Fields Related to a Z-File, page 168
- Implode Fields, page 169
- Cross Reference Fields, page 169
- Fields with Verification, page 169
- Fields with no Verification, page 169

Standard retrieval functions are described in Chapter **Retrieval** in the *Predict Reference Manual*.

Fields and Related Views (Code R)

Reports on fields and the related fields in related files. *Related file* means a master file and its userview. The relationship between fields is established as described below depending on whether the view is derived from a single master file or from several master files.

- **Single-Master Views**
Views and userviews derived from a **single** master file, for example an Adabas C file and its userview, are related by field short name (see page 116).
- **Multiple-Master Views**
For views which can be derived from **several** master files (SQL tables and views), the coupling is established by parameters *from Table/View ID* and *from Field ID* in the field List of the file documenting the view.

Command: RELATED ELEMENT

See **Rippling**, page 264, in Chapter **File** for more information on related fields and files.

Non-Standard Fields (Code N)

Lists fields which are not derived from standard files, and also fields which were derived from standard files but subsequently changed to non-Standard fields.

Command: NONSTANDARD ELEMENT

Fields Related to a Z-File (Code Z)

Reports on fields which are derived from standard files.

Command: STANDARD ELEMENT

Implode Fields (Code *I*)

Lists objects related to fields in the form of a structured list (see below).

The following information is given:

- the files to which the fields belong
- databases to which those files belong
- programs using those files
- programs containing those programs
- the systems to which those programs belong
- the master files to the views

```

15:58:19          ***** P R E D I C T 4.1.1 *****          1999-11-11
                                - Implode Field -          Page:      1
Field ID ..... DEMO-IMPL-EL

File ID ..... DEMO-IMPL-FI ,Type=A
File Relation ID ... DEMO-IMPL-RL ,Type=D
  Database ID ..... DEMO-IMPL-DA ,Type=A
  Program ID ..... DEMO-IMPL-PR1 ,Type=P
    System ID ..... DEMO-IMPL-SY ,Type=A
***** End of Report *****

```

Command: IMPLODE ELEMENT

Fields with Verification (Code *T*)

Reports on all fields that have a verification.

Command: CHILDREN EL VE

Cross Reference Fields (Code *X*)

Lists all objects that are linked to the file containing the field.

Command: XREF ELEMENT

Fields with no Verification (Code *U*)

Reports on all fields that do not have a verification.

Command: EMPTY EL VE

Layout of Field Lists

- Two different list formats are used for displaying information on fields:
- when fields are listed without entering a specific file (format 1 below)
 - when all fields used in a specific file are listed (format 2)

Format 1

The first list format applies when fields of several files are listed.

08:58:18		***** P R E D I C T 4.1.1 *****				1999-11-17
		- List Field -				Page: 1
Cnt	Ty L	Field ID	F	Length	D	File ID
1	1	FH-001	A	50.0		PD-ADA-LONG
2	1	FH-002	A	50.0		PD-ADA-LONG
3	1	FH-003	A	50.0		PD-ADA-LONG

Meaning of Columns

Ty	Type of field. See page 98 for a complete list of Field types and codes. <i>RE</i> indicates a redefinition.
L	The field level. Level number of the field. See page 99.
Field ID	ID of the field object.
F	The field format. See page 100.
Length	The field length. See page 102.
D	Descriptor type. See page 113.
File ID	ID of the file containing the field.

Format 2

The second list format is used when all fields used in a specific file are listed.

14:44:34

***** P R E D I C T 4.1.1 *****

1999-08-29

- List Field -

File ID ARH-A1

Type ADABAS C file

Fnr 12

Cnt	Ty	L	Field ID	F	Length	D	U	DB	S	Occ
1		1	FIELD1	A	5.0			AA	N	
2		1	FIELD2	A	3.0			AB	N	
3		1	FIELD3	A	4.0			AC	N	
4		1	FIELD4	A	6.0			AD	N	
5	RE	1	FIELD4							
6		2	FIELD4-RE1	A	3.0					
7		2	FIELD4-RE2	A	3.0					

Meaning of Columns

U	Unique option. <i>U</i> is displayed if the field is a unique descriptor.
DB	Field short name. See page 116.
S	Suppression / Null Value option. See page 117.
Occ	Number of occurrences for multiple fields. See page 115.

Output Options for Field Retrieval

Retrieval Type	D				N				Z				T				U			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																				
Adabas sizes																				
Association attributes																				
Attributes	✓				✓				✓				✓				✓			
Check expression																				
Composed Fields	✓				✓				✓				✓				✓			
Connecting character													✓	✓						
Cover page	✓		✓		✓		✓		✓		✓		✓	✓	✓	✓	✓		✓	
Description	✓				✓				✓				✓	✓			✓			
Display length			✓				✓				✓				✓				✓	
Display modifier	✓				✓				✓				✓				✓			
Dummy/Placeholder														✓		✓				
DV-Field expression	✓				✓				✓				✓				✓			
Entry points																				
Extract														✓						
Generation layout																				
Adabas version																				
Language																				
Alignment/sync.																				
Position/Offset																				
Counter length																				
Compiler																				
Replace with syn.																				

Retrieval Type	D				N				Z				T				U			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓				✓				✓				✓	✓			✓			
Linked Verification																				
Mark implementation														✓						
No. abstract lines	✓		✓		✓		✓		✓		✓		✓	✓	✓	✓	✓		✓	
Natural options	✓				✓				✓				✓				✓			
Owner	✓				✓				✓				✓	✓			✓			
With users	✓				✓				✓				✓	✓			✓			
Page size (<i>only in batch or printout</i>)	✓		✓		✓		✓		✓		✓		✓	✓	✓	✓	✓		✓	
Procedure code																				
Rules																				
Show implementation																				
Sorted by Field	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
Subquery																				
Synonyms	✓		✓		✓		✓		✓		✓		✓		✓		✓		✓	
STARTAB elements																				
Trigger																				
Use Con-form	✓				✓				✓				✓	✓			✓			
User exit	✓				✓				✓				✓	✓			✓			
3GL specification	✓				✓				✓				✓				✓			

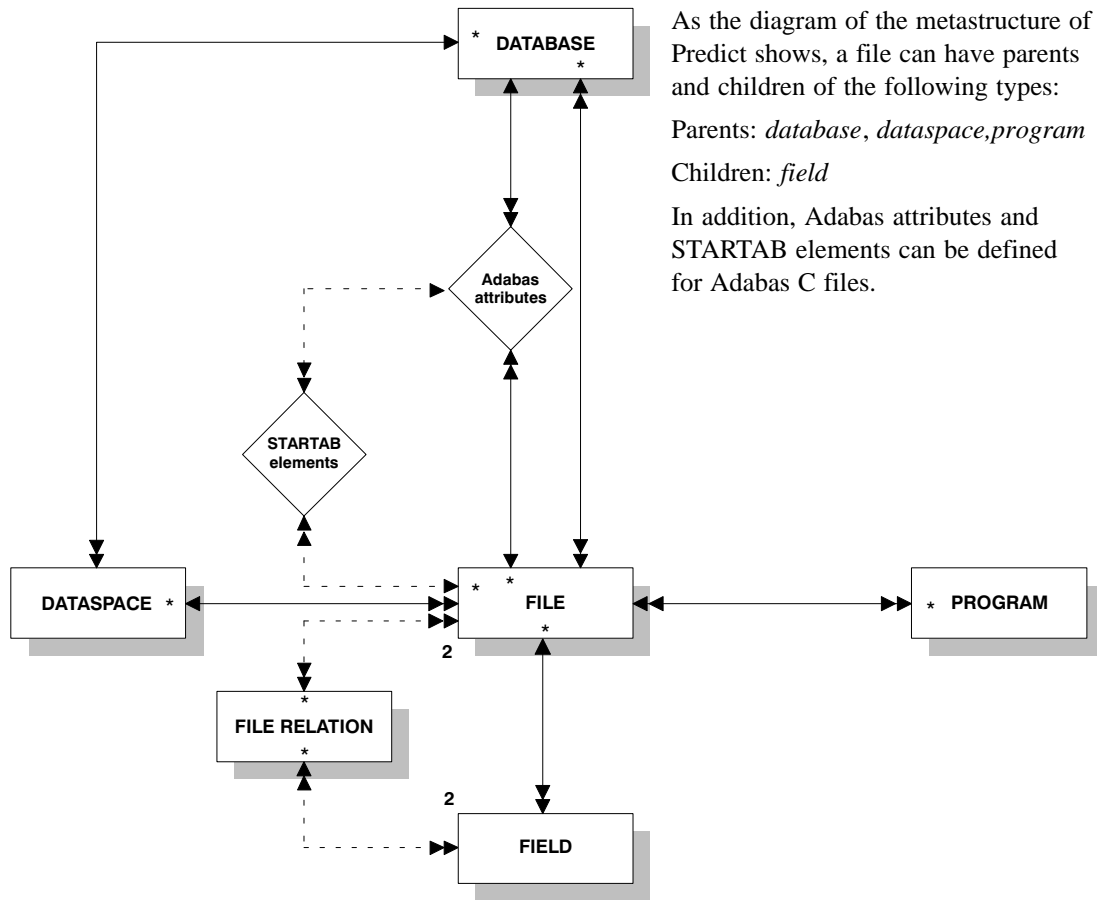
Output Options for Field Retrieval (Continued)

Retrieval Type	I				X				R			
Output Mode	T				X				L			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes												
Adabas sizes												
Association attributes												
Attributes						✓						
Check expression												
Composed Fields									✓			
Connecting character	✓					✓						
Cover page	✓				✓	✓			✓			
Description						✓						
Display length									✓			
Display modifier												
Dummy/Placeholder						✓						
DV-Field expression												
Entry points												
Extract						✓						
Generation layout												
Adabas version												
Language												
Alignment/sync.												
Position/Offset												
Counter length												
Compiler												
Replace with syn.												

Retrieval Type	I				X				R			
Output Mode	T				X				L			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r
Keywords						✓						
Linked Verification												
Mark implementation						✓			✓			
No. abstract lines	✓					✓			✓			
Natural options												
Owner						✓						
With users						✓						
Page size (<i>only in batch or printout</i>)	✓				✓	✓			✓			
Procedure code												
Rules												
Show implementation												
Sorted by Field									✓			
Subquery												
Synonyms									✓			
STARTAB elements												
Trigger												
Use Con-form						✓						
User exit												
3GL specification												

FILE

With Predict objects of type *File*, file structures can be defined for a wide variety of data storage systems and for use with different programming languages.



How this Chapter is Organized

- **The File Maintenance Menu**, page 179
 - Common file attributes, page 182
- **Documenting Files of Different Types**, page 187
 - Adabas C files and userviews, page 187
 - File types *Conceptual*, *Standard* and *Other*, page 199
 - SQL file Types, page 200
 - Adabas SQL, page 211
 - Adabas Cluster Table, page 213
 - Adabas D, page 216
 - DB2, page 219
 - ORACLE, page 231
 - INGRES, page 227
 - INFORMIX, page 223
 - SYBASE, page 236
 - General SQL file, page 239
 - rdb, page 241
 - IMS, page 242
 - VSAM, page 246
 - ISAM, page 250
 - Entire System Server, page 252
- **File Maintenance** page 254
 - Purge file, page 255
 - Rename/renumber file, page 256
 - Edit list of fields, page 257
 - Force standard, page 260
 - Push backward, page 260
 - Modify Adabas attributes, page 263
 - Modify STARTAB elements, page 263
 - Edit subquery of a file, page 263

- **Rippling – Ensuring Consistent Data Definitions** page 264
 - Overview, page 265
 - Rippling from standard files to master files, page 268
 - Rippling from master files to views/userviews, page 272
- **File Retrieval**, page 278
 - The file retrieval screen, page 279
 - File-specific retrieval parameters, page 279
 - File-specific retrieval functions
 - Difference of files, page 280
 - List files related to a file, page 282
 - Layout of file lists, page 283
 - Output options for file retrieval, page 285

Where to Find Detailed Information on Defining Distributed Data Structures

Basic information on attributes of files and how to execute file-specific functions is given in the sections below. If your aim is to define data in the good old fashioned way (using simple files in a database not accessible via a network), you will find all the required information in the sections below. Knowledge needed additionally when defining complex data distribution structures using Adabas Star or Entire Transaction can be found in the respective chapters of the Manual *Predict and Other Systems*.

The File Maintenance Menu

The *File Maintenance Menu* is displayed with function code *M* and object code *FI* in a Predict *Main Menu* or with the command MAINTAIN FILE.

```
15:07:27          ***** P R E D I C T  4.1.1  *****          1999-01-23
Plan  10          - (FI) File Maintenance -          Profile JCA

Function                                Function
A  Add a file                          L  Link children
C  Copy file                          O  Edit owners of a file
D  Display file                       S  Select file from a list
M  Modify file                        B  Push backward
N  Rename/Renumber file              F  Force standard
P  Purge file                        J  Modify ADABAS attributes
W  Edit description of a file         K  Modify STARTAB elements
                                      Y  Edit subquery of a file

Function .....
File ID .....
Copy ID .....
External name ..
in database ....
Restrictions ..*   Profile JCA ,used          Child type .....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next
```

Parameters

Function

All standard maintenance functions are described in Chapter **Maintenance** in the *Predict Reference Manual*.

The function *Edit list of Fields* does not appear on the menu but can be called with function code *E*.

The functions *Add a File* and *Modify File* can be called with the editor line command *.E* when editing a file list (no matter which type of object the file list belongs to). See Chapter **Editors in Predict** in the *Predict Reference Manual* for more information. The following file maintenance functions are described later in this chapter:

Parameters

	<p>Purge file, page 255</p> <p>Rename/Renumber file, page 256</p> <p>Edit list of fields, page 257</p> <p>Force standard, page 260</p> <p>Push backward, page 260</p> <p>Modify Adabas attributes, page 263</p> <p>Modify STARTAB elements, page 263</p> <p>Edit subquery of a file, page 263</p>
File ID	<p>For the <i>Select</i> function:</p> <p>specifies a file ID to be used as a selection criterion, either alone or in combination with parameters <i>Files of type</i> and <i>in database</i>. If this field is left blank, all files which satisfy other selection restrictions specified are listed.</p> <p>See naming conventions for individual file types on page 182.</p>
File of type	<p>For the <i>Select</i> function:</p> <p>a file type can be specified as an additional selection criterion.</p> <p>For the <i>Add</i> and <i>Copy</i> functions:</p> <p>if file type is specified here, it will be passed to the <i>Add/Copy File</i> screen.</p> <p>Enter an asterisk to display a selection window with the file types valid for a particular function in your environment.</p> <p>See complete list of valid file types on page 182.</p>
Copy ID	<p>Identifies the target file ID for the functions <i>Copy</i> and <i>Push backward</i>.</p> <p>For function <i>Push backward</i>: the ID of a standard file (type Z).</p>
External name	<p>For the <i>Select</i> function:</p> <p>name of the file in another environment. Up to 50 characters can be specified here.</p> <p>Up to 250 characters can be specified with the <i>Modify file</i> function. If <i>External name</i> exceeds 50 characters, enter Y in the <i>Zoom</i> field.</p>

Parameters

in database	<p>For the <i>Select</i> function: a database ID can be specified as an additional selection criterion. Asterisk notation is possible.</p> <p>For the <i>Add</i> and <i>Copy</i> functions: the database ID can be specified here. This ID will be passed to the <i>Add/Copy file</i> screen.</p> <p>See list of valid database and file types on page 183.</p>
File nr	<p>For the <i>Select</i> function: A file number can be specified as an additional selection criterion.</p> <p>For the <i>Add</i> and <i>Copy</i> functions: The file number can be specified here. This number is passed to the <i>Add a file</i> or <i>Copy file</i> screen.</p>
Restrictions	<p>Additional criteria can be specified to restrict the scope of files to be processed. See Restrictions page 81 in Chapter The User Interface in the Manual <i>Introduction to Predict</i>.</p>
Child type	<p>For function <i>Link children</i>: Objects of this type are to be linked to the file. Valid values: <i>Field</i> (default) or via user-defined association to any other object type.</p>

Common File Attributes

The following attributes are applicable to all or most file types.

File ID

For naming conventions valid for all object types see page 6.

Special naming conventions apply to SQL file types. See overview in section **SQL File Types** on page 202.

File Type

A file object has one of the following types. The file type must be compatible with the database in which it is contained. See table on page 183.

A	Adabas C file	U	Adabas C userview
AT	Adabas Cluster Table	V	VSAM file (physical)
B	Adabas C SQL view	W	Physical VSAM view
C	Conceptual file	X	General SQL file
D	DB2 table	Z	Standard file
E	DB2 view	1	LEASY
F	rdb file	2	ISAM BS2000
I	IMS segment	OT	ORACLE table
J	IMS segment layout	OV	ORACLE view
K	IMS userview	BT	Adabas D table
L	Logical VSAM file	BV	Adabas D view
M	ISAM file	JT	INGRES table
O	Other	JV	INGRES view
P	Entire System Server file	YT	SYBASE table
Q	Entire System Server userview	YV	SYBASE view
R	Logical VSAM view	XT	INFORMIX table
S	SEQUENTIAL file	XV	INFORMIX view
T	RMS file		

In Database

The ID of the database containing the file. The database type must be compatible with the file type.

To generate a DDM for a file, the file must be linked to a compatible database (not of type C).

File Type		Compatible Database Type	
A	Adabas C File	A	Adabas C Database
A(SQL)	Adabas C File with <i>SQL usage</i> set to <i>Y</i>	Q	Adabas SQL Handler
AT	Adabas Cluster Table		
B	Adabas C SQL view		
BT	Adabas D table	B	Adabas D Handler
BV	Adabas D view		
D	DB2 table	D	DB2 Database
F	rdb File	R	RDB Handler
I	IMS segment	I	IMS Database
JT	INGRES table	J	INGRES Handler
JV	INGRES view		
L	Logical VSAM File	V	VSAM Handler
OT	ORACLE table	O	ORACLE Handler
OV	ORACLE view		
P	Entire System Server File	P	Entire System Server
T	RMS File	M	RMS Handler
V	Physical VSAM File	V	VSAM Handler
X	General SQL File	E	General SQL Handler
XT	INFORMIX table	X	INFORMIX Handler
XV	INFORMIX view		
YT	SYBASE table	Y	SYBASE Handler
YV	SYBASE view		
1	LEASY	H	Other Handler
2	ISAM BS2000		
All File Types		C	Conceptual

File number

The number of the file. The possible value depends on the file type:

File Type	File Number
AT, J, K, Q, R, U	File number is taken from the specified master file
B, D, E, I, X, BT, BV, JT, JV, OT, OV, XT, XV, YT, YV	<i>not applicable</i>
Other file types	1 - 5000

Note:

The file number can only be changed with the function *Rename/Renumber File*.

Natural Construct Parameters

The following parameters are only relevant if you are using Natural Construct. They appear in every *Add*, *Copy* or *Modify file* screen.

Literal name	String to be used by Natural Construct in messages issued to confirm (un)successful access of a file via a DDM generated from the Predict file object.
Average count	The average number of records contained in the file.
Stability	Indicates how permanent the data contained in the file is.
	F Fixed. The file contains information which will always be valid, for example days of the week.
	S Stable. The file contains information which does not change very often, for example file EMPLOYEES.
	V Volatile. The file contains information which is constantly being updated, for example an invoice file.
	<i>blank</i> Not specified (default value).

Defining Basic File Attributes

The following screen is displayed for the *Add a File* and *Copy File* functions for all file types:

15:05:04

***** P R E D I C T 4.1.1 *****

1999-01-23

- Add a file -

File ID JCA-NEW3

File type* C Conceptual file

Master file*

File number*

logical ADASTAR type ...*

in database*

General Parameters

- File type
- The file type. Enter an asterisk for list of possible values or see list on page 182.
- Master file
- For file types listed below, enter the ID of the related file. The type of related file is given in the following table:

File Type	Type of Master File	
AT	A	Adabas C File
J and K	I	IMS segment
L and W	V	Physical VSAM File
Q	P	Entire System Server File
R	L	Logical VSAM File
U	A	Adabas C File

-
- The master file can be selected using asterisk notation.
- File number
- See table of possible values on page 184.

General Parameters

Logical ADASTAR type	How the logical file is to be stored:
E	Expanded
P	Partitioned
R	Replicated
N	PROPAGATOR file. Not applicable when defining data distribution for Adabas Star.
<i>blank</i>	Simple file (default).

Note: This parameter is only applicable to files of type *Adabas C*. For files of other types, this parameter must be *blank*.

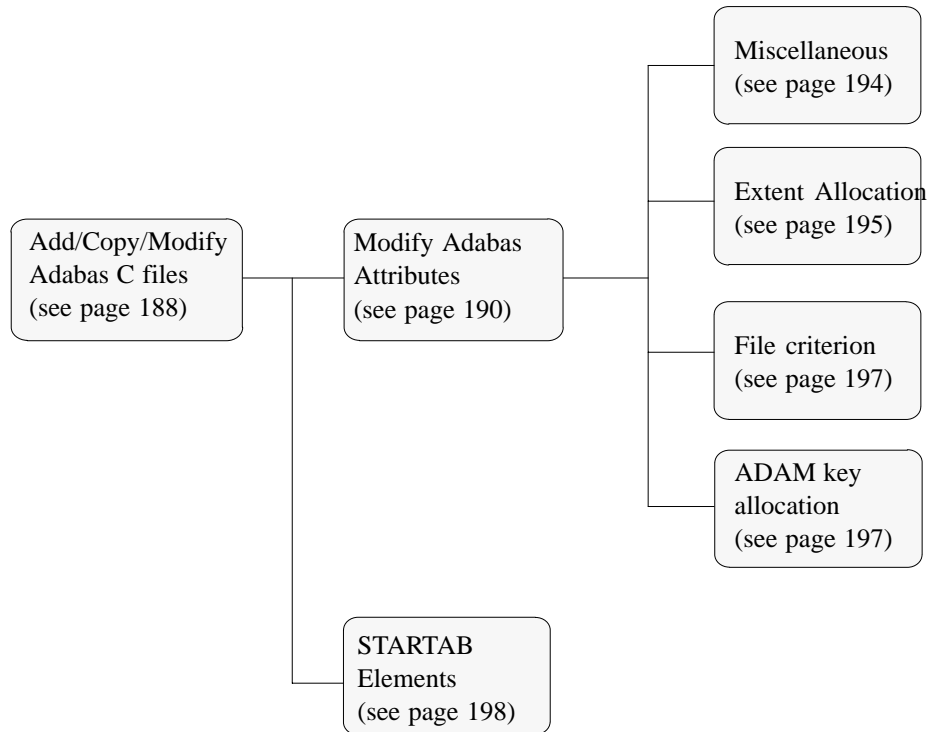
Documenting Files of Different Types

Adabas C Files, File Type A

Adabas C files are defined in several subsequent input screens.

Screens on lower levels are called by specifying *Y* in the field *MORE attributes* in the EDIT line of the higher-level screens.

The following diagram gives an overview of the input screens and the sections where these screens are described.



The Add/Copy/Modify File Screen

09:28:21

***** P R E D I C T 4.1.1 *****

1999-04-26

- Add a file -

File ID jca-a-file

Type ADABAS C, Simple file

File number 123

in database

Keys ..

Zoom: N

Literal name

Average count

Stability* Not specified

Sequence field*

ADASTAR access number ..* 0 L-DBnr .. L-Fnr ..

ADABAS C SQL usage N (Y/N)

Abstract Zoom: N

EDIT: Owner: N Desc: N Field list: N MORE Attr.: Y

Note:
Parameters common to all object types are described under **Global Attributes**, page 6.
For parameters common to all file types see **Common File Attributes**, page 182.

Parameters

Sequence field	<p>The descriptor to be used by Natural for logical sequential reading.</p> <p>Determines the sequence in which records are delivered by the READ LOGICAL statement.</p> <p>The GENERATE DDM function will use this field as the default READ LOGICAL field in the Natural data definition module.</p>
----------------	--

ADASTAR access number, L-DBnr, L-Fnr

The logical database number and the logical file number are derived from the *ADASTAR access number* (and vice-versa). The *L-DBnr* and *L-Fnr* are used as database and file number for function *Generate DDM* if the parameter *Use ADASTAR access-nr* is set to *Y* or *T* in the *Generate DDM* menu. Valid values are 257 to 65279. No check for uniqueness is performed.

Note: This parameter should not be confused with the STARTAB parameter *ADASTAR number*, which is used to identify a file uniquely within a network. See **Including the Definition in the StarTAB Table** in Chapter **Adabas Star** in the Manual *Predict and Other Systems*

Adabas C SQL usage

Y File is accessible via Adabas SQL Server.

Note: When you add a file, this parameter can be specified in the *Add a file* screen. To change the value of this attribute for a file that already exists, use the *Rename/renumber file* function (see page 256).

Additional Options in the EDIT Line

MORE Attr.

Y Two types of additional attributes can be specified:

- Adabas attributes
- STARTAB elements.

The screens for entering Adabas attributes are described in the sections below.

Modifying Adabas Attributes

There are different ways of calling the initial *Modify Adabas attributes* screen:

- specify *Y* in the field *MORE attributes* in the EDIT line and mark *Adabas attributes* in selection window
- select function *Modify Adabas Attributes* function (code *J*) in the *File Maintenance* menu
- enter command *.A* in the file editor of a database object
- enter command *MODIFY ADA-ATTR.*

```

16:19:12          ***** P R E D I C T  4.1.1  *****          1999-01-23
                        - Modify ADABAS attributes -
File ID ..... JCA-NEW3                      Added 1998-01-23 at 16:19
Type ..... ADABAS, Simple file                by JCA
in database ....

Required attributes                               Phys. ADASTAR type
Phys. file number ..* 123                        Simple file
Min ISN ..... 1
Max ISN .....

      Device      Cylinder Blocks      Padding factor      Max 2. alloc
      *-----
ASSO      3390      UI                      10
           NI
DATA      3390      DS                      10

Loading attributes                               Loading attributes
Max recl. ....
ISN reusage ..... N (Y,N)                      One AC extent ..... N (Y,N)
User ISN ..... N (Y,N)                        DS reusage ..... Y (Y,N)
Mixed DS device ..... N (Y,N)

EDIT:   Owner: N   Desc: N   Field list: N   MORE:   Attributes: N

```

Note:

Up to five additional input screens can be called from this screen (see also diagram on page 187).

Parameters

- Required attributes
- Phys. file number

If a database is specified, the file number is taken as a physical file number automatically if this is possible. If not, a free physical number can be selected from a selection window.
- Physical ADASTAR type

The ADASTAR type of the physical file which describes how the logical file is stored. Read only field.
The types for the physical file are limited by the logical ADASTAR type, as shown in the following table:

Physical ADASTAR Type		Logical ADASTAR Type		Remarks
E	expanded	E	expanded	
P	partitioned	P	partitioned	
PM	partitioned master	P	partitioned	Any replicated partition (type <i>PR</i>) of a partitioned logical File (type <i>P</i>) must have an associated partitioned master File (type <i>PM</i>).
PR	partitioned replicated	P	partitioned	
R	replicated	R	replicated	
RM	replicated master	R		Any replicated logical File (type <i>R</i>) can have at most one physical File of type <i>RM</i> ; its other physical Files must have type <i>R</i> .
leer	simple File	any		

- Min ISN

ADALOD LOAD parameter MINISN.
- Max ISN

ADALOD LOAD parameter MAXISN.

Device and Size Specification for Adabas C Files

```

.....
.....
Device      Cylinder Blocks  Padding factor  Max 2. alloc
*-----
ASSO      3380    UI                      10
           NI
DATA      3380    DS                      10
.....
.....

```

The device type and the size of the Upper Index (UI), Normal Index (NI) and Data Storage (DS) can be specified. If the size is specified in blocks, the equivalent size in cylinders provided by Predict is preceded by *greater than* (>) if the number of cylinders does not match exactly. If the size is specified in cylinders, Predict provides the equivalent size in blocks.

The maximum secondary allocation in blocks can also be specified in each case.

Four characters specify the type of device used to store this part of the file. This device type must already be defined in the Predict database object containing this file. When this device type is changed in the database, the same change should be made in every file object contained in the database.

DATA padding factor	ADALOD LOAD parameter DATAPFAC.
ASSO padding factor	ADALOD LOAD parameter ASSOPFAC.
Device	The device type of the Upper Index (<i>UI</i>), Normal Index (<i>NI</i>) and Data Storage (<i>DS</i>). The device type for Data Storage is ADALOD LOAD parameter DSDEV.
Size (Cylinders/Blocks)	ADALOD LOAD parameters UISIZE (Upper Index), NISIZE (Normal Index) and DSSIZE (Data Storage).

Note:

See also **Extent allocation**, page 195.

Loading attributes

Max recl.	ADALOD LOAD parameter MAXRECL.
ISN reusage	ADALOD LOAD parameter ISNREUSE.
User ISN	ADALOD LOAD parameter USERISN.
One AC extent	ADALOD LOAD parameter NOACEXTENSION.
DS reusage	ADALOD LOAD parameter DSREUSE.
Maximum secondary allocation	ADALOD LOAD parameters MAXUI (Upper Index), MAXNI (Normal Index) and MAXDS (Data Storage).

Additional Options in the EDIT Line

MORE Attributes	Y	Displays a window for specifying the following Adabas attributes: <ul style="list-style-type: none">- ADASTAR or PROPAGATOR attributes- Miscellaneous attributes- ADAM key definition- Extent allocation- File criterion
-----------------	---	--

Note:
ADASTAR or PROPAGATOR attributes and *Extent allocation* only appear in this window if applicable.

Miscellaneous Attributes

```

16:33:18          ***** P R E D I C T  4.1.1  *****          1999-08-23
                        - Modify ADABAS attributes -
File ID ..... JCA-NEW3          Added 1998-01-23 at 16:19
in database .....                by JCA
PDBnr .....          PFnr ... 123

ADABAS Security definition
  Access level ..... (0-15)
  Update level ..... (0-15)

Loading attributes
  Ciphred ..... N (Y,N)
  Mirror ..... N (Y,N)
  LOWNERID ..... 0 (0-8)
  Refresh from program ... N (Y,N)
  Automatic allocation ... Y (Y,N)
  PLOG ..... Y (Y,N)
  ISN size .....* 0
  Erase ..... Y (Y,N)

```

Parameters

Access level	The Adabas access security level of the file.
Update level	The Adabas update security level of the file.
Ciphred	Y The file is a ciphred file.
Mirror	Y The file is reflective in this database and therefore included in the file list for the ADAREF parameter file.
LOWNERID	Parameter used in Adabas C Version 5.3 and above.
Refresh from program	Adabas C parameter PGMREFRESH. See <i>Adabas DBA Reference Manual</i> .
Automatic allocation	Y Adabas C will automatically allocate and deallocate extents. See <i>Adabas Reference Manual</i> .
PLOG	Y Database runs with protection log. UNIX only.
ISN Size	Length of ISN. Valid values: 0, 2, 3 and 4. For Adabas/UNIX: 0, 2 and 4 are valid. For mainframes: 0, 3 and 4 are valid.

Erase Y For Adabas/UNIX. All index and data storage blocks are overwritten with zeroes when they are returned to the the free space table.

Extent Allocation (Size Specifications For More Than One Extent)

More than one extent can be specified using the *Extent allocation* option in the *Modify Adabas attributes* selection window.

The size and first RABN (Start Rb) of the Address Converter (AC), Upper Index (UI), Normal Index (NI) and Data Storage (DS) can be specified for up to five extents. The total space allocated is displayed in the lower right corner of the screen.

```
16:47:50          ***** P R E D I C T  4.1.1  *****          1999-01-23
                    - Modify ADABAS attributes -
File ID ..... HEB-FI-PART          Modified 1998-07-20 at 09:24
in database .... HEB-DA-TRANS          by FST
PDBnr ..... 21      PFnr ... 1

Ext      *Dev Start Rb Cylinder Blocks          *Dev Start Rb Cylinder Blocks
1. AC 3390                      1          2.
   UI 3390
   NI 3390
   DS 3390

3. AC                      4.
   UI
   NI
   DS

5. AC          +- Total --- Cylinder - Blocks ---+
   UI          ! AC                      1          !
   NI          ! UI                      !
   DS          ! NI                      !
               ! DS                      !
               +-----+
EDIT:      Owner: N      Desc: N      Field list: N
```

Specifying Restrictions on Input Data (File Criteria)

File criteria determine which data can be written to a file.

```
11:16:39          ***** P R E D I C T  4.1.1  *****          1999-10-21
                        - Modify ADABAS attributes -
File ID ..... PD-AD1                      Modified 1998-10-21 at 10:08
in database ..... DEMO-DB                      by PD
PDBnr ..... 180   PFnr ... 13

Ty Distribution key          F Length  Occ  D U DB N NAT-1
-- *-----
  FELDD                      A  12.00          XC N

      1 Operator ..* EQ  Value format ..* A  Length .. 2
      Value ..... AB
                                           Zoom: N

Connected via ..* O
      2 Operator ..* EQ  Value format ..* A  Length .. 12
      Value ..... ac
                                           Zoom: N

Connected via ..*
      3 Operator ..*      Value format ..*      Length ..
      Value .....
                                           Zoom: N
                                           Scroll to:

EDIT:   Owner: N   Desc: N * Field list: Y
```

Parameters

Distribution key	ID of the field to be checked. Format, length, number of occurrences, descriptor type, uniqueness option, short name, suppression option and Natural length are displayed. The field must exist in the file.
Operator	EQ equal LT less LE less equal GT greater GE greater equal
Value format	Format of the specified value. Can differ from the field format. Valid values can be selected from selection window.

Length	Length to be evaluated.
Value	Value to be checked. If the value is longer than 40 characters, set <i>Zoom</i> to <i>Y</i> .
Scroll to	If more validation criteria are specified than can be displayed in one screen, the criteria to be displayed on top of the list can be specified in the field <i>Scroll to</i> .

Modifying ADAM Descriptor Definition

```
16:40:40          ***** P R E D I C T  4.1.1  *****          1999-01-23
                        - Modify ADABAS attributes -
File ID ..... JCA-NEW3                               Added 1998-01-23 at 16:34
in database .....                                     by JCA
PDBnr .....          PFnr ... 123

ADAM descriptor definition
Field ID .....*
Parm .....
Overflow .....
```

Parameters

ADAM descriptor definition

Field ID	Fields to be used as ADAM descriptor. ADALOD LOAD parameter ADAMDE.
Parm	ADALOD LOAD parameter ADAMPARM.
Overflow	ADALOD LOAD parameter ADAMOFLOW.

Modifying STARTAB Elements

```
12:51:47          ***** P R E D I C T   4.1.1 *****          1999-07-01
                                - Add STARTAB element -
File ID ..... JPE340                                Added 1998-07-01 at 12:51
Type ..... ADABAS, Simple File                        Modified

Network .....*
User ID .....*
ADASTAR File number ..... L-DBnr .. 0   L-Fnr .. 0
Simple ..... Y (Y,N)

                                Database      PDBnr   PFnr   phys.
                                *-----*      *-----*   ADASTAR type
                                -----*-----*   -----*
                                1
```

Parameters

See section **Including the Definition in the StarTAB Table** in Chapter **Adabas Star** in the Manual *Predict and Other Systems* for a description of all parameters.

File Types Conceptual, Standard and Other

```
10:36:36          ***** P R E D I C T  4.1.1  *****          1999-06-30
                                - Add a file -
File ID ..... FI
Type ..... Conceptual file
File number .....
in database .....
Keys ..
Zoom: N

Literal name .....
Average count .....
Stability .....*   Not specified
Sequence field .....*
Abstract      Zoom: N
```

Note:
Parameters *in database* and *Sequence field* do not apply to files of type *Standard* and *Other*.

Parameters

File ID	The ID of the file object.
Type	The file type.
File number	Files of these types can have a file number from 0 – 99999.
in database	The ID of the database containing the file. See page 183.
Sequence field	The function <i>Generate DDM</i> will use this field as the default READ LOGICAL field in the Natural data definition module. For conceptual files for documentation and later use.
Literal name, Average count, Stability	These three parameters are only applicable if you are using Natural Construct. See page 184.

SQL File Types

Predict offers various file types for documenting tables and views of the SQL systems listed below. The file objects which document the SQL tables and views can be used to generate SQL CREATE statements, DDMs and copy code members for 3GLs. The CREATE statements are stored as Natural members in file FDIC.

How this Section is Organized

- **General Information**
The following information applies to file objects documenting the SQL systems listed below. Type-specific information is given in the respective sections starting from page 210.
 - Naming Conventions for SQL file Types, page 201
 - Common Parameters for SQL file Types, page 203
 - Adding fields to the field Lists of SQL Views, page 205
- **Documenting SQL Tables and View of Different Types** page 210
 - Adabas C SQL view, page 211
 - Adabas D, page 216
 - DB2, page 219
 - ORACLE, page 231
 - INGRES, page 227
 - INFORMIX, page 223
 - SYBASE, page 236

Naming Conventions for SQL Objects

Special naming conventions apply to the following objects in Predict

- SQL file types. See table below.
- Fields linked as children to these file types
- Constraint names
- Correlation names
- Tablespace for ORACLE
- The file IDs must be fully qualified.

A fully qualified ID consists of three parts:

 - Hyphen to separate creator/schema from table/view name
 - Table/view name. The maximum length depends on the SQL system. See table below.
- Fully qualified IDs may not exceed 32 characters.
- The permitted characters listed in the table below apply to creator/schema and table/view name.

Convention	File Type	AT,B A(SQL)	BT, BV	D, E	JT, JV	OT, OV	X	XT, XV	YT, YV
Maximum length of table/view name		32	18	18	24	30	18	18	30
Upper case				✓		✓	✓		
Upper/lower case		✓	✓		✓			✓	✓
'_' allowed at first pos.				✓	✓				✓
'#' allowed at first pos.			✓	✓					
'\$' allowed at first pos.			✓	✓					
'@' allowed at first pos.			✓	✓					
'_' allowed from second pos.		✓	✓	✓	✓	✓	✓	✓	✓
'#' allowed from second pos.			✓	✓	✓	✓	✓		✓
'\$' allowed from second pos.			✓	✓	✓	✓	✓		✓
'@' allowed from sec. pos.			✓	✓	✓				✓
Numbers allowed from second pos.		✓	✓	✓	✓	✓	✓	✓	✓

Type-specific rules are also given in the respective sections of this chapter.

Common Parameters for SQL File Types

The following parameters are valid for all or most SQL file types.

SQL Attributes

These parameters apply to all SQL views.

Select	A	Select all: Redundant duplicates are not eliminated.
	D	Select distinct: Redundant duplicates are eliminated.
With check option	Y	All inserts and updates to the view are checked against the view definition.

Edit Line Options

Profile options are described in Chapter **Defaults** in the *Predict Administration Manual*. The editors are described in Chapter **Editors in Predict** in the *Predict Reference Manual*.

EDIT Subquery

This option is available for all SQL views.

Enter *Y* in the *EDIT Subquery* field to call an Editor to edit the subquery clause of the SQL view. The editor called depends on the preferences specified in the *Profile > Handling* screen:

- if your first choice editor is *Natural*, the Subquery Editor (a modified Natural Editor) is called.
- if your first choice editor is *SAG* or *Word for Windows*, the SAG Editor is called.

Additional commands are available for processing subqueries and checks are performed when the subquery is cataloged.

See Chapter **Editors in Predict** in the *Predict Reference Manual*.

EDIT Trigger

This option is available for the following SQL tables:

- Adabas D
- DB2
- ORACLE
- Informix
- INGRES
- SYBASE

It is also available for the following file type:

- General SQL file

Enter *Y* in the *EDIT Trigger* field to edit the check expression or trigger(s) of the file. The editor called depends on the preferences specified in the *Profile > Handling* screen:

- if your first choice editor is *Natural*, the Description Editor (a modified Natural Editor) is called.
- if your first choice editor is *SAG* or *Word for Windows*, the SAG Editor is called.

No special checks are performed when triggers or check expressions are saved.

Field Lists of SQL Views

The following screen shows the layout of the field list of an SQL file.

>			> + Fi: ARH-E1		L: 1 S: 5	
Ty	L	Field ID	from Table/View ID		Field ID	All
*_	-	-----	-----		-----	
	1	ARH1	ARH-D1		ARH1	
SP	1	ARH_SP	ARH-D1		ARH_SP	
	1	ARH4	ARH-D1		ARH4	
	1	TIME_1	PD-E1		TIME_1	

Column	Meaning
Ty	Field type.
L	Field level.
Field ID	ID of field object documenting the SQL view. The ID of the field object in Predict documenting a field in a view can differ from the name of the field in the original table or view.
from Table/View ID	ID of the Predict file documenting the table or view from which the field was taken. If this file contains a subquery clause with a correlation name for the table or view, the correlation name must be entered instead of the file ID.
from field ID	Field in the table or view from which it was taken.

Adding new Fields to Field Lists of SQL Views

New fields can easily be inserted into the field list of an SQL view using one the following two methods:

Manually

Enter parameters *Field ID*, *from Table/View ID* and *from Field ID* described above. See naming conventions for SQL objects on page 202.

With Command SELECT

Use the command SELECT to select fields from other SQL tables or views and insert them into the current field list. The following screen appears:

```

15:06:46          ***** P R E D I C T  4.1.1  *****          1999-05-02
Plan    2          - Field Selection Menu -                      Profile JCA

File ID ..... JCA-XV                                           Added 1998-04-27 at 16:25
                                                                by JCA

Select object type ..... EL  ( Field )

Retrieval type .....* D
Output mode .....* S Select

Search criteria
  Field ID/Synonym ...                                           Synonym of language*
  in file .....                                                in files of type ..*

Restrictions .....*      Profile JCA ,used

```

Note:

Parameters not listed here are described in Chapter **Retrieval** in the *Predict Reference Manual*.

Parameters

Retrieval type	The following field retrieval functions are available to select fields for insertion into the field list: D Fields N Non-standard fields U Fields with no verification.
Search criteria	
Field ID/Synonym	This parameter corresponds to <i>from Field ID</i> in the field list of SQL views shown above.
in file	This parameter corresponds to <i>from Table/View ID</i> in the field list of SQL views shown above.
in files of type	It is possible to select fields of any type for insertion. However, an error message will be given when you try and catalog a field list containing fields with an incompatible type. See list of compatible field types on page 209. If a unique field ID is specified, this parameter is ignored.
Mark the fields to be inserted with X, S or /. Selected fields are marked <i>*ins*</i> in the field list. Catalog the list to add the fields to the list.	

Editing the Subquery of an SQL View

Calling the Editor

Two methods are available for calling an editor to edit the subquery of an SQL view:

- enter *Y* in the EDIT *subquery* field in the bottom line of the *Add File*, *Copy File* or *Modify File* screen, or
- enter the command EDIT FILE SUBQUERY <*File ID*>

The editor called depends on the preferences specified in the *Profile > Handling* screen:

- if your first choice editor is *Natural*, the Subquery Editor (a modified Natural Editor) is called.
- if your first choice editor is *SAG* or *Word for Windows*, the SAG Editor is called.

```
>                                     > + VI: PD-E1                                L: 1      S: 8
All  ....+....1....+....2..Subquery clause .4....+....5....+....6....+....7..
      FROM
      PD-D1 D1 ,
      SMR-D
      WHERE
        'ABC' IN
          ( SELECT A-COL2 FROM PD-D1 A)
```

Structure of a Subquery Clause

The following rules apply:

- In the first part of the subquery clause, the related master files and their correlation names can be specified in SQL syntax.
- The file type of the related master files must be compatible with the file type:

File Type of View	Related Master File Type
B	A(SQL), AT, B
BV	BT, BV
E	D, E
JV	JT, JV
OV	OT, OV
XV	XT, XV
YV	YT, YV

- Any correlation name that is specified must be used whenever the file is referred to. Type-dependent rules apply to the length of a correlation name and the characters permitted. See page 202.
- The first part of the subquery is generated automatically if the fields of the file are defined in Predict before the subquery is edited.
- The second part of the subquery contains the selection criteria of the view: the WHERE clause, GROUP BY clause or HAVING clause or any combination of these. The name of each field referenced in the selection criteria must be qualified by the ID of the file from which the field is taken or – if a correlation name has been specified in the first part of the subquery – by the correlation name.
- When generating a CREATE VIEW statement for a view, hyphens (-) are replaced by underscores (_) or points (.).
- The subquery can include comment lines (with /*, * or ** in the first two columns) and line comments (preceded by /*).

Documenting SQL Tables and Views of Different Types

Tables and view of the following SQL systems can be documented in Predict:

- Adabas SQL, page 211
- Adabas Cluster Table, page 213
- Adabas D, page 216
- DB2, page 219
- INFORMIX, page 223
- INGRES, page 227
- ORACLE, page 231
- SYBASE, page 236

Adabas SQL Server

How this Section is Organized

- Overview
- Naming Conventions
- Adabas Cluster Table, page 213
- Adabas C SQL view, page 215

Overview

There are two methods of documenting Adabas tables:

- **Files of Type A (SQL)**
If an Adabas table corresponds **exactly** to a base table in Adabas SQL Server, it can be documented as a file of type *A (SQL)*. The Adabas file must not contain groups structures or multiple value fields. Rotated fields are not supported with this method. This method is retained for reasons of compatibility with earlier Predict versions.
- **Files of Type AT**
Tables can also be documented with files of type *AT* (Adabas cluster table). Files of this type can be understood as userviews to an Adabas file. See page 213.

Adabas SQL **views** are documented with files of type *B*. See page 215.

Naming Conventions

The following naming conventions apply to files documenting Adabas SQL Server tables and views (files of type *AT*, *B*).

Upper / lower case

If the parameter *General Defaults > Miscellaneous > Upper/lower case / Object ID* is set to *L*, the following attributes are stored in upper and lower case as entered:

- File ID (object IDs containing lower case letters are not recommended)
- Triggers
- Derived field expressions

- SQL verifications
- Check expressions
- Constraint names

See also Chapter **Defaults** in the *Predict Administration Manual*.

Length

Table/View names for Adabas SQL Server objects can have up to 32 characters.

Permitted characters

See overview of permitted characters on page 202.

Qualifier

The identifier of a table or view must be given in qualified form: the schema identifier, a delimiter and the table/view name. A hyphen is used as a delimiter (not a period as in SQL). An example: SYSSAG-SYSCOLUMNS. Hyphens in names are treated as follows:

- When a table/view is generated from a Predict file object, the hyphen is transformed into a period (.).
- Because hyphens are used as delimiters, only one hyphen can occur in the SQL identifier. Column names must not contain a hyphen.
- The hyphen can be used as a minus sign or negative sign in the field expression or the subselect clause and must then be preceded by a blank.

Adabas Cluster Table

```
11:25:05          ***** P R E D I C T 4.1.1 *****          1999-06-06
                        - Add a file -

File ID ..... JCA-AT
Type ..... ADABAS cluster table
File number ..... 1234 Master file: JCA-A
in database .....
Keys ..
Zoom: N

Literal name .....
Average count .....
Stability .....* Not specified
ADASTAR access number ..* 0      L-DBnr ..      L-Fnr ..
Table level .....*

Abstract      Zoom: N

EDIT:  Owner: N   Desc: N   Field list: N
```

Note:
Parameters not listed below are described in other sections of this manual:
Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.
Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.
See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	See naming conventions on page 202.
in database	ID of the database object containing the file.
Table level	0 Only “flat” structures are permitted (no MU or PE fields). 1 For defining multiple fields and periodic groups. 2 For defining multiple fields within a periodic group.

Parameters

There are two methods of documenting periodic groups and multiple value fields in AT files:

- If the occurrences of PE/MU fields are **fixed**, you can use rotated fields in the AT file.
- If the occurrences of PE/MU fields are **variable**, use subtables (AT files at level 1 or level 2).

For more information see Chapter **Adabas SQL Server** in the Manual *Predict and Other Systems*.

Adabas C SQL View

15:24:04

***** P R E D I C T 4.1.1 *****

1999-04-26

- Add a file -

File ID JCA-FIB1

Type ADABAS C SQL view

in database

Keys ..

Zoom: N

Literal name

Average count

Stability* Not specified

SQL attributes

Select* A

With check option N (Y/N)

Abstract

Zoom: N

EDIT: Owner: N Desc: N Field list: N Subquery: N

Note:
Parameters not listed below are described in other sections of this manual:
Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.
Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.
See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	See naming conventions on page 202.
in database	ID of the database object containing the file.

Adabas D

Adabas D tables and views can be documented in Predict with file objects of type *BT* and *BV* respectively. These file objects can be used to generate DDMs or CREATE TABLE/VIEW statements.

Naming Conventions

The following naming conventions apply to files documenting Adabas D tables and views.

Upper / lower case

If the parameter *General Defaults > Miscellaneous > Upper/lower case / Object ID* is set to *L*, the following attributes of Adabas D objects are stored in upper and lower case as entered:

- File ID (object IDs containing lower case letters are not recommended)
- Triggers
- Derived field expressions
- SQL verifications
- Check expressions
- Constraint names

See also Chapter **Defaults** in the *Predict Administration Manual*.

Length

- Table/View names for Adabas D objects can have up to 18 characters.
- A fully qualified ID (*Creator + Hyphen + Table/View name*) may not exceed 27 characters.

Permitted characters

See overview of permitted characters on page 202.

Adabas D Table, File Type *BT*

```
14:49:52          ***** P R E D I C T 4.1.1 *****          1999-04-26
                        - Modify file -
File ID ..... JCA-BT                      Modified 1998-03-24 at 14:23
Type ..... ADABAS D table                  by JCA
in database .....
Keys ..                                     Zoom: N

Literal name .....
Average count .....
Stability .....*   Not specified
Check constraint name
Abstract      Zoom: N

EDIT:   Owner: N * Desc: N   Field list: N   Trigger: N
```

Note:
Parameters not listed below are described in other sections of this manual:
Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.
Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.
See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	See naming conventions for Adabas D objects on page 216.
in database	ID of the database object containing the file. To generate a DDM from files of type <i>Adabas D table</i> , the file must be linked to a database of type <i>Adabas D handler</i> .
Check constraint name	If a table check expression has been defined and the name of a check constraint is entered here, the following clause is generated in the CREATE TABLE statement: CONSTRAINT <i>constraint_name</i> CHECK (<i>check_expression</i>)

Adabas D View, File Type *BV*

14:36:40

***** P R E D I C T 4.1.1 *****

1999-04-26

- Add a file -

File ID JCA-BV

Type ADABAS D view

in database

Keys ..

Zoom: N

Literal name

Average count

Stability* Not specified

SQL attributes

Select* A

With check option N (Y/N)

Abstract

Zoom: N

EDIT:

Owner: N

Desc: N

Field list: N

Subquery: N

Note:
Parameters not listed below are described in other sections of this manual:
Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.
Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.
See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	See naming conventions for Adabas D objects on page 216.
in database	ID of the database object containing the file. To generate a DDM from files of type <i>Adabas D view</i> , the file must be linked to a database of type <i>Adabas D handler</i> .

DB2

DB2 tables and views can be documented in Predict with file objects of type *D* and *E* respectively. These file objects can be used to generate DDMs or CREATE TABLE/VIEW statements.

How this Section is Organized

- Naming Conventions, page 219
- DB2 table, page 220
- DB2 view, page 222

Naming Conventions

The following naming conventions apply to files documenting DB2 tables and views.

Upper / lower case

File IDs must be entered in upper case. If the parameter *General Defaults > Miscellaneous > Upper/lower case / Object ID* is set to *L*, lower-case IDs are not converted to upper case and an error message is given.

Hyphens

- A hyphen is used to delimit the creator from the table/view name.
- Only one hyphen is permitted in the ID of a DB2 table/view object.
- When a table or view is generated from the Predict file object, the hyphen is converted to a period.

Length

- Table/View names for DB2 objects can have up to 18 characters.
- A fully qualified ID (*Creator + Hyphen + Table/View name*) must not exceed 27 characters.

Permitted characters

See overview of permitted characters on page 202.

DB2 Table, File Type *D*

```

14:36:40          ***** P R E D I C T  4.1.1  *****          1999-04-26

  File ID ..... JCA-DB2                      Added 1999-07-24 at 15:34
  Type ..... DB2 table                        by JCA
  in database .....
  Keys ..                                     Zoom: N

  Literal name .....
  Average count .....
  Stability .....*   Not specified
  DB2 attributes
    Number of partitions ..                  CCSID .....* A ASCII
    Edit program .....                      Temporary ...* N (Y/N)
    Validation program ....
    Audit .....* N
    OBid .....
    Data capture ..... N (Y/N)
    Restrict on drop ..... (Y/N)
    Check constraint name .
  Abstract      Zoom: N

EDIT:   Owner: N   Desc: N   Field list: N   Trigger: N

```

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	ID of the Predict object documenting the DB2 table. See naming conventions on page 219.
in database	ID of the database object containing the file.
DB2 Attributes	
Number of partitions	The number of partitions of the table.
Edit program	The name of an edit routine for the table.
Validation program	The name of a validation routine for the table.
Audit	The type of access to this table that will cause auditing to be performed. Valid values: A All C Changes N None
OBid	Identifies the OBID to be used for the table. An OBID is the identifier for an object's internal descriptor in DB2.
	<i>Note:</i> This parameter is required if parameter <i>DB2 ROSHARE parm</i> of the database object containing the table is set to <i>R</i> . See page 24.
	See your DB2 documentation for more information.
Data capture	Y Data changes are passed to a user exit.
Restrict on drop	Y The DB2 table cannot be dropped. To drop a table with this setting, this parameter must be set explicitly to <i>N</i> .
Check constraint name	If a table check expression has been defined and the name of a check constraint is entered here, the following clause is generated in the CREATE TABLE statement: CONSTRAINT <i>constraint_name</i> CHECK (<i>check_expression</i>)
CCSID	Encoding scheme. Valid values: <i>blank</i> not specified A ASCII E EBCDIC

Temporary	Y	Global temporary table
	N	not temporary.
Edit: Trigger		See page 204.

DB2 View, File Type *E*

```

11:28:33          ***** P R E D I C T 4.1.1 ***** 1999-04-27
                    - Modify file -

File ID ..... JCA-E                                Added 1998-04-20 at 13:44
Type ..... DB2 view                                by JCA
in database ..... B-ARH-DA-C
Keys ..                                             Zoom: N

Literal name .....
Average count .....
Stability .....*      Not specified
SQL attributes
  Select .....* A
  With check option ..... N (Y/N)

Abstract      Zoom: N

EDIT:   Owner: N   Desc: N * Field list: N   Subquery: N

```

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	ID of the Predict object documenting the DB2 view.
in database	ID of the database object containing the file.

INFORMIX

INFORMIX tables and views can be documented in Predict with file objects of type *XT* and *XV* respectively. These file objects can be used to generate DDMs or CREATE TABLE/VIEW statements.

How this Section is Organized

- Naming Conventions, page 223
- INFORMIX Table file Type *XT*, page 224
- INFORMIX View, file Type *XV*, page 226

Naming Conventions

The following naming conventions apply to files documenting INFORMIX tables and views.

Upper / lower case

If the parameter *General Defaults > Miscellaneous > Upper/lower case / Object ID* is set to *L*, the following attributes of INFORMIX objects are stored in upper and lower case as entered:

- File ID (object IDs containing lower case letters are not recommended)
- Triggers
- DV field expressions
- SQL verifications
- Check expressions
- Constraint names

See also Chapter **Defaults** in the *Predict Administration Manual*.

Length

- Table/View names for INFORMIX objects can have up to 18 characters.
- A fully qualified ID (*Creator + Hyphen + Table/View name*) may not exceed 27 characters.

Permitted characters

- IDs containing special characters must be enclosed in double quotes, for example: "USR1"-"FIL£ABC".
- See overview of permitted characters on page 202.

INFORMIX Table, File Type *XT*

```

17:14:58          ***** P R E D I C T  4.1.1  *****          1999-04-26
                        - Modify file -
File ID ..... JCA-XT                      Modified 1998-04-27 at 10:49
Type ..... INFORMIX table                  by JCA
In database .....
Keys ..                                     Zoom: N

Literal name ....
Average count ...
Stability .....*   Not specified
Informix ONLINE . N (Y/N)
Extensize .....
Nextsize .....
Lock mode .....*
DBspace/Path.
Abstract      Zoom: N

```

```

EDIT:   Owner: N   Desc: N   Field list: N   Trigger: N

```

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	ID of the Predict object documenting the INFORMIX table. See naming conventions for INFORMIX objects on page 223.
in database	ID of the database object containing the file. To generate a DDM from files of type <i>INFORMIX table</i> , the file must be linked to a database of type <i>INFORMIX Handler</i> .
INFORMIX ONLINE	Y An INFORMIX ONLINE database is used.
<i>Note:</i> The following parameters are only applicable if INFORMIX ONLINE is set to Y.	
Extensize	Size of the initial extent for the table and and its key.
Nextsize	Size of subsequent extents which are added if necessary.
Lock mode	Determines whether locking is set to page level or row level. P Page level locking. R Row level locking.
DBspace/Path	Name of the DBspace where INFORMIX ONLINE is to store the table. If this parameter is not specified, the table is stored in the DBspace of the database entered under <i>in database</i> .

INFORMIX View, File Type XV

17:15:37

***** P R E D I C T 4.1.1 *****

1999-04-26

- Add a file -

File ID JCA-XV

Type INFORMIX view

in database

Keys ..

Zoom: N

Literal name

Average count

Stability* Not specified

SQL attributes

Select* A

With check option N (Y/N)

Abstract

Zoom: N

EDIT:

Owner: N

Desc: N

Field list: N

Subquery: N

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	ID of the Predict object documenting the INFORMIX view. See naming conventions for INFORMIX objects on page 223.
in database	ID of the database object containing the file. To generate a DDM from files of type <i>INFORMIX view</i> , the file must be linked to a database of type <i>INFORMIX Handler</i> .

INGRES

INGRES tables and views can be documented in Predict with file objects of type *JT* and *JV* respectively. These file objects can be used to generate DDMs or CREATE TABLE/VIEW statements.

How this Section is Organized

- Naming Conventions, page 227
- INGRES Table, file Type *JT*, page 228
- INGRES View, file Type *JV*, page 230

Naming Conventions

The following naming conventions apply to files documenting INGRES tables and views.

Upper / lower case

If the parameter *General Defaults > Miscellaneous > Upper/lower case / Object ID* is set to *L*, the following attributes of INGRES objects are stored in upper and lower case as entered:

- File ID (object IDs containing lower case letters are not recommended)
- Triggers
- DV field expressions
- SQL verifications
- Check expressions
- Constraint names

See also Chapter **Defaults** in the *Predict Administration Manual*.

Length

- Table/View names for INGRES objects can have up to 24 characters.
- A fully qualified ID (*Creator + Hyphen + Table/View name*) may not exceed 32 characters.

Permitted characters

See overview of permitted characters on page 202.

INGRES Table, File Type *JT*

```

17:13:01          ***** P R E D I C T  4.1.1  *****          1999-04-26
                                - Modify file -
File ID ..... JCA-JT                      Added 1998-04-20 at 10:28
Type ..... INGRES table                    by JCA
In database .....
Keys ..                                     Zoom: N

Literal name .....
Average count .....
Stability .....*   Not specified
Journaling ..... Y (Y/N)
Duplicated ..... Y (Y/N)
Abstract      Zoom: N

```

```

EDIT:   Owner: N   Desc: N   Field list: N   Trigger: N

```

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	ID of the Predict object documenting the INGRES table. See naming conventions for INGRES objects on page 227.
in database	ID of the database object containing the file. To generate a DDM from files of type <i>INGRES table</i> , the file must be linked to a database of type <i>INGRES Handler</i> .
Journaling	Y The clause WITH JOURNALING is entered in the CREATE statement. N The clause WITH NO JOURNALING is entered in the CREATE statement.
Duplicated	Y The clause WITH DUPLICATES is entered in the CREATE statement. N The clause WITH NO DUPLICATES is entered in the CREATE statement.

INGRES View, File Type JV

17:13:50

***** P R E D I C T 4.1.1 *****

1999-04-26

- Add a file -

File ID JCA-JV

Type INGRES view

in database

Keys ..

Zoom: N

Literal name

Average count

Stability* Not specified

SQL attributes

Select* A

With check option N (Y/N)

Abstract

Zoom: N

EDIT: Owner: N Desc: N Field list: N Subquery: N

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	ID of the Predict object documenting the INGRES view. See naming conventions for INGRES objects on page 227.
in database	ID of the database object containing the file. To generate a DDM from files of type <i>INGRES view</i> , the file must be linked to a database of type <i>INGRES Handler</i> .

ORACLE

ORACLE tables and views can be documented in Predict with file objects of type *OT* and *OV* respectively. These file objects can be used to generate DDMs or CREATE TABLE/VIEW statements.

How this Section is Organized

- Naming Conventions, page 231
- ORACLE Table, file Type *OT*, page 232
- ORACLE View file Type *OV*, page 235

Naming Conventions

The following naming conventions apply for ORACLE objects (Files of type *OT* and *OV*)

Upper / lower case

IDs must be entered in upper case. If the parameter *General Defaults > Miscellaneous > Upper/lower case / Object ID* is set to *L* and you try and enter a file ID containing lower case letters, an error message is given.

See also Chapter **Defaults** in the *Predict Administration Manual*.

Length

- Table/View names for ORACLE objects can have up to 30 characters.
- A fully qualified ID (*Creator + Hyphen + Table/View name*) must not exceed 32 characters.

Permitted characters

- IDs containing special characters must be enclosed in double quotes, for example: "USR1"-"FILEABC".
- See overview of permitted characters on page 202.

ORACLE Table, File Type *OT*

```

14:33:56          ***** P R E D I C T  4.1.1  *****          1999-04-26
                                - Modify file -
File ID ..... JCA-OT                      Modified 1998-04-20 at 12:43
Type ..... ORACLE table                      by JCA
In database .....
Keys ..                                         Zoom: N

Literal name ....
Average count ...
Stability .....*   Not specified
Pctfree .....
Initrans .....      Pctused ....
Tablesapce name .      Maxtrans ...
Cluster name ....
Cluster column .*
Check constraint name ..
Storage clause
  Initial .....      Next .....
  Minextents ....      Maxextents .
  Pctincrease ...
Abstract      Zoom: N

EDIT:   Owner: N * Desc: N * Field list: N   Trigger: N

```

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	ID of the Predict object documenting the ORACLE table. See naming conventions for ORACLE objects on page 231.
in database	ID of the database object containing the file. To generate a DDM from files of type <i>ORACLE table</i> , the file must be linked to a database of type <i>ORACLE Handler</i> .
Pctfree	If an integer from 1 – 99 is specified here, the clause PCTFREE <i>n</i> is generated in the CREATE TABLE statement. PCTFREE reserves a set amount of room in every block allocated to a table for future updates to that table's data.
Pctused	If an integer from 1 – 99 is specified here, the clause PCTUSED <i>n</i> is generated in the CREATE TABLE statement. PCTUSED specifies the minimum level of space usage that ORACLE will maintain for each block of the table.
Initrans	If a value from 1 – 255 is entered here, the clause INITRANS <i>n</i> is generated in the CREATE TABLE statement. INITRANS is the initial number of transaction entries that are allocated within each block.
Maxtrans	If a value from 1 – 255 is entered here, the clause MAXTRANS <i>n</i> is generated in the CREATE TABLE statement. MAXTRANS specifies the maximum number of transactions that may update a data block concurrently.
Tablespace name	If a tablespace name is entered here, the clause TABLESPACE <i>name</i> is generated in the CREATE TABLE statement. This name represents the tablespace in which the table will be created.
Cluster name	If a cluster name is entered here, the clause CLUSTER <i>name</i> is generated in the CREATE TABLE statement. The table is to be included in the specified cluster.
Cluster column	Table columns that correspond to the cluster columns of the cluster specified under <i>Cluster name</i> .

Check constraint name	If a table check expression has been defined and the name of a check constraint is entered here, the following clause is generated in the CREATE TABLE statement: <code>CONSTRAINT <i>constraint_name</i></code> <code>CHECK (<i>check_expression</i>)</code>
Storage clause	If specified, the values below are used in the STORAGE clause generated with the CREATE TABLE statement. All of the values below must be specified as integers.
Initial	The size in bytes of the first extent allocated when the object is created – the original amount of space allocated to the object.
Next	The size in bytes of every subsequent extent to be allocated.
Minextents	The total number of extents to be allocated when the segment is created.
Maxextents	The total number of extents, including the first, which can ever be allocated.
Pctincrease	The percent by which each NEXT extent will grow over the last extent allocated.

See your ORACLE documentation for more information on these ORACLE-specific parameters.

ORACLE View, File Type *OV*

14:35:07

***** P R E D I C T 4.1.1 *****

1999-04-26

- Modify file -

File ID JCA-OV

Type ORACLE view

in database

Keys ..

Literal name

Average count

Stability* Not specified

SQL attributes

Select* A

With check option N (Y/N)

Check constraint name ..

Abstract Zoom: N

Modified 1998-04-20 at 10:10

by JCA

Zoom: N

EDIT: Owner: N * Desc: N Field list: N Subquery: N

Note:
Parameters not listed below are described in other sections of this manual:
Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.
Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.
See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	ID of the Predict object documenting the ORACLE view. See naming conventions for ORACLE objects on page 231.
in database	ID of the database object containing the file. To generate a DDM from files of type <i>ORACLE table</i> , the file must be linked to a database of type <i>ORACLE Handler</i> .
Check constraint name	Name of check option used if parameter <i>With check option</i> is set to <i>Y</i> . See page 203.

SYBASE

SYBASE tables and views can be documented in Predict with file objects of type *YT* and *YV* respectively. These file objects can be used to generate DDMs or CREATE TABLE/VIEW statements.

How this Section is Organized

- Naming Conventions, page 236
- SYBASE Table, file Type *XT*, page 237
- SYBASE View, file Type *XV*, page 238

Naming Conventions

The following naming conventions apply to files documenting SYBASE tables and views.

Upper / lower case

If the parameter *General Defaults > Miscellaneous > Upper/lower case / Object ID* is set to *L*, the following attributes of SYBASE objects are stored in upper and lower case as entered:

- File ID (object IDs containing lower case letters are not recommended)
- Triggers
- DV field expressions
- SQL verifications
- Check expressions
- Constraint names

See also Chapter **Defaults** in the *Predict Administration Manual*.

Length

- Table/View names for SYBASE objects can have up to 30 characters.
- A fully qualified ID (*Creator + Hyphen + Table/View name*) must not exceed 32 characters.

Permitted characters

- IDs containing special characters must be enclosed in double quotes, for example: "USR1"-"FIL£ABC".
- See overview of permitted characters on page 202.

SYBASE Table, File Type *YT*

17:18:12

***** P R E D I C T 4.1.1 *****

1999-04-26

- Add a file -

File ID JCA-YT

Type SYBASE table

In database

Keys ..

Literal name

Average count ...

Stability* Not specified

Database name ...

Segment name

Abstract Zoom: N

Zoom: N

EDIT: Owner: N Desc: N Field list: N Trigger: N

Note:
Parameters not listed below are described in other sections of this manual:
Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.
Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.
See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	ID of the Predict object documenting the SYBASE table. See naming conventions for SYBASE objects on page 236.
in database	ID of the database object containing the file. To generate a DDM from files of type <i>SYBASE table</i> , the file must be linked to a database of type <i>SYBASE Handler</i> .
Database name	Name of the database in SYBASE containing the table.
Segment name	Name of the segment where the table is to be placed

SYBASE View, File Type YV

17:19:57

***** P R E D I C T 4.1.1 *****

1999-04-26

- Add a file -

File ID JCA-YV

Type SYBASE view

in database

Keys ..

Zoom: N

Literal name

Average count

Stability* Not specified

SQL attributes

Select* A

With check option N (Y/N)

Abstract

Zoom: N

EDIT:

Owner: N

Desc: N

Field list: N

Subquery: N

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

See also **Common Parameters for SQL File Types** on page 203.

Parameters

File ID	ID of the Predict object documenting the SYBASE table. See naming conventions for SYBASE objects on page 236.
in database	ID of the database object containing the file. To generate a DDM from files of type <i>SYBASE table</i> , the file must be linked to a database of type <i>SYBASE Handler</i> .

General SQL File, File Type X

Files of type *General SQL File* are used to document all SQL systems not explicitly supported by Predict.

16:10:04

***** P R E D I C T 4.1.1 *****

1999-08-23

- Modify file -

File ID JCA-X

Type General SQL file

in database

Keys ..

Literal name

Average count

Stability*

Check constraint name ..

Abstract Zoom: N

Added 1998-04-20 at 12:51

by JCA

Zoom: N

EDIT: Owner: N Desc: N * Field list: N Trigger: N

Note:
Parameters not listed below are described in other sections of this manual:
Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.
Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

Parameters

File ID	ID of the Predict object
in database	ID of the database object containing the file. To generate a DDM from files of type <i>General SQL file</i> , the file must be linked to a database of type <i>General SQL handler</i> .
Check constraint name	The name of a check constraint can be entered here.
EDIT: Trigger	Y The editor called to edit the check expression of the file depends on your setting in the <i>Profile > Handling</i> screen. See page 204.

rdb

```
12:27:16          ***** P R E D I C T  4.1.1  *****          1999-05-09
                        - Add a file -

File ID ..... JCA-RDB
Type ..... rdb file
File number .... 123
in database .....
Keys ..                                           Zoom: N

Literal name .....
Average count .....
Stability .....*   Not specified
Sequence field .....*
Abstract      Zoom: N
```

Note:
Parameters not listed below are described in other sections of this manual:
Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.
Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

Parameters

File ID	ID of the file object.
in database	ID of the database containing the file (see page 183).
Sequence field	The descriptor to be used by Natural for logical sequential reading. Determines the sequence in which records are delivered by the READ LOGICAL statement. The GENERATE DDM function will use this field as the default READ LOGICAL field in the Natural data definition module.

IMS

IMS Segment Layouts and Userviews (File Types *J* and *K*)

```

11:14:40          ***** P R E D I C T 4.1.1 *****          1999-06-30
                        - Add a file -
File ID ..... JCA-J
Type ..... IMS seg. layout
File number ..... IMS segment: CHD-ARTCHD-ART
in database .....
Keys ..
Zoom: N

Literal name ....
Average count ...
Stability .....* Not specified
IMS attributes
  Segment name .. ART          Parent ....
  min. length ...             Source-1 ..
  max. length ... 32000        Source-2 ..
  Segment type ..
Abstract      Zoom: N

EDIT:  Owner: N   Desc: N   Field list: N

```

The following attributes of an IMS segment (type *I*) are shown for that file and for the related files of types *J* and *K*.

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

Parameters

File number	The number of the file. A read only field. The number of the related IMS segment is shown. See page 184 for more information.
IMS attributes	
Segment name	The name of the IMS segment from which the related Predict file object of type <i>I</i> was incorporated.
Min. length	The minimum length of the IMS segment (zero if the length is fixed).
Max. length	The maximum length of the IMS segment (if it is fixed).
Segment type	<p>The type of the IMS segment. Possible values:</p> <p>Logical child (<i>C</i>) Logical (<i>L</i>) Physical (<i>P</i>) Virtual (logical) child (<i>V</i>).</p> <p>Segments of type <i>logical</i> occur only in logical IMS databases. Segments of types <i>child</i>, <i>physical</i> and <i>virtual</i> occur only in physical IMS databases.</p>
Parent	The ID of the Predict file object of type <i>I</i> incorporated from the parent segment of the IMS segment (the segment one level above it in the hierarchical structure of the IMS database). For a root segment, this field is left blank.
Source-1	<p>The following rules apply:</p> <ul style="list-style-type: none">- For a segment of type <i>V</i>, the ID of the Predict file object of type <i>I</i> that was incorporated from the related segment of type <i>C</i>.- For a segment of type <i>L</i>, the ID of the Predict file object of type <i>I</i> that was incorporated from the segment of a physical database from which this segment of a logical database is derived.- For a segment of type <i>CHILD</i> or <i>P</i>, this field is left blank.

Parameters

Source-2	<p>The following rules apply:</p> <ul style="list-style-type: none">- For a segment of type <i>LOGICAL</i> derived from a segment of type <i>C</i>, the ID of the Predict file object of type <i>I</i> that was incorporated from the logical parent of the segment of type <i>C</i>.- For a segment of type <i>LOGICAL</i> derived from a segment of type <i>V</i>, the ID of the Predict file object of type <i>I</i> that was incorporated from the logical parent of the segment of type <i>V</i> (the physical parent of the related segment of type <i>C</i>).- For any other segment, this field is left blank.
----------	--

Editing Field Lists of IMS Files

Restrictions that apply when editing a field list of an IMS file depend on the type of the IMS file and are described in the table on the next page.

File Type	Restrictions
I (IMS Segment)	<p>The following attributes can be maintained: <i>ID</i>, <i>keywords</i>, <i>owners</i>, <i>abstract</i>, <i>format</i>, <i>NAT hdr1-3</i> (Natural headers), <i>NAT editm</i> (Natural edit mask), <i>3GL specification</i>, <i>Condition name & value</i> and <i>Field name synonyms</i>.</p> <p>See Defining Basic Attributes of Fields, page 95, and Defining More Attributes of fields, page 139, in Chapter Field in the Manual <i>Predefined Object Types in Predict</i>.</p> <p>No fields can be added or deleted. Format changes are rippled across related files of type <i>J</i> or <i>K</i>. Only the following changes of format are allowed:</p> <ul style="list-style-type: none">- between P (packed) and PS (packed signed);- between P6 or P7 and D (date);- between P12 or P13 and T (time).

File Type	Restrictions
J (IMS Segment Layout)	<p>The following rules apply:</p> <ul style="list-style-type: none"> - A file of type <i>J</i> can contain user-defined fields and fields of the related file of type <i>I</i>. The two-character short names of the user-defined fields must fall within the range preceding the parameter <i>Start in logical</i> defined by the DDA in the <i>Miscellaneous</i> defaults of the <i>Modify General Defaults</i> function. Its value is normally HA. - Fields of the related file of type <i>I</i> that are included in a File of type <i>J</i> must have the same attributes in the File of type <i>J</i> as they have in the file of type <i>I</i>. - Their offset in the file of type <i>J</i> must be the same as their IMS-OFFSET in the file of type <i>I</i>. <p>For a variable-length segment, only one field in one file of type <i>J</i> can be defined as variable length.</p> <ul style="list-style-type: none"> - If it is a field, it must be the last field in the segment. - If it is a multiple value field or a periodic group, it can be anywhere in the segment. - However, if it is not the last field, its maximum occurrence must be specified. <p>Predict checks that the above conditions are met when the field list of the file is cataloged. Changes to user-defined fields are rippled across related files of type <i>J</i> or <i>K</i>.</p>
K (IMS Userview)	<p>A file of type <i>K</i> can contain fields of the related file of type <i>J</i> and fields of all related files of type <i>J</i>. <i>ID, keywords, owners, comments, format, NAT hdr1-3</i> (Natural headers) and <i>NAT editm</i> (Natural edit mask), <i>3GL specification, Condition name & value</i> and <i>Field name synonyms</i> can be maintained.</p>

VSAM

The following sections contain the following:

- Physical VSAM file (file type *V*)
- VSAM logical files, VSAM userviews (file types *L*, *W* and *R*)

See also Chapter **VSAM** in the Manual *Predict and Other Systems*.

Physical VSAM File (File Type *V*)

```

16:38:48          ***** P R E D I C T 4.1.1 *****          1999-05-09
                        - Add a file -

File ID ..... JCA-VMS
Type ..... VSAM file
File number ..... 123
in database .....
Keys ..
                                           Zoom: N

Literal name .....
Average count .....
Stability .....*   Not specified
Sequence field .....*

VSAM attributes          Location          Data set attributes
VSAM DD name .....      Volume 1 ..      CI size
VSAM file org .....* K KSDS      Volume 2 ..      Data .....
Compressed file .... N (Y/N)      Volume 3 ..      Index .....
Numeric zones .....* F           Volume 4 ..      Recsize
                                           Min .....
                                           Max .....
                                           Free space .. %

Abstract      Zoom: N

EDIT:   Owner: N   Desc: N   Field list: N

```

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

Parameters

Sequence field	<p>The descriptor to be used by Natural for logical sequential reading.</p> <p>Determines the sequence in which records are delivered by the READ LOGICAL statement.</p> <p>The function <i>Generate DDM</i> uses this attribute as the default READ LOGICAL field in the Natural data definition module.</p>
VSAM attributes	
VSAM DD name	<p>This parameter refers to a DD card in batch mode, or to a CICS FCT object.</p> <p>See <i>Natural VSAM Installation Notes</i> and the <i>Natural 2 Operations Manual</i>.</p>
VSAM file org	<p>Valid values:</p> <p>K KSDS (key-sequenced data set)</p> <p>E ESDS (entry-sequenced data set)</p> <p>R RRDS (relative-record data set)</p>
Compressed file	<p>Only applicable to files with organization <i>K</i> (KSDS).</p> <p>Y The record will be truncated if the trailing byte positions are unused.</p>
Numeric zones	<p>Valid entries are <i>C</i> and <i>F</i>. This field affects the representation of positive numbers in packed decimal format. The sign position holds hexadecimal <i>C</i> or <i>F</i> respectively.</p>
Location	
Volume 1 – 5	<p>The volume(s) on which the file is located. Up to five volumes can be specified.</p>
Dataset attributes	
CI size data	The data control interval size.
CI size index	The control interval size for the primary index.
RECSIZE min	The minimum record size.
RECSIZE max	The maximum record size.
Free space	The free space to be allocated (in percent).

VSAM Logical Files, VSAM Userviews (File Types *L*, *W* and *R*)

```

11:48:33          ***** P R E D I C T 4.1.1 *****          1999-06-30
                        - Add a file -

File ID ..... JCA-L
Type ..... Logical VSAM
File number ..... 1
in database .....
Keys ..
                                           Zoom: N

Literal name .....
Average count .....
Stability .....*   Not specified

VSAM attributes
  VSAM prefix .....
  Sequence field .....*
  Organisation ..... KSDS
  Related ..... ARH-VSAM
Abstract      Zoom: N

EDIT:   Owner: N   Desc: N   Field list: N

```

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

Parameters

VSAM prefix	<p>Only applicable to files of types <i>L</i> and <i>R</i>.</p> <p>If this field is left blank, the last 3 digits of the file number are taken as the prefix. Otherwise, a string of up to 20 characters can be specified. The records in the corresponding physical VSAM file (type <i>V</i>) whose primary keys begin with the specified prefix string will be considered as belonging to the logical VSAM file. The length of the primary key specified for the logical VSAM file must be equal to the length of the primary key specified for the physical VSAM file <i>minus</i> the length of the prefix.</p> <p>A dummy field (corresponding to the prefix) preceding the primary key in the logical VSAM file must be defined for the field offsets to be calculated correctly.</p>
Org	<p>The organization of the parent physical VSAM file (type <i>V</i>):</p> <p>Valid values:</p> <ul style="list-style-type: none">K KSDS (key-sequenced data set)E ESDS (entry-sequenced data set)R RRDS (relative-record data set)
Related	<p>The ID of the related physical VSAM file (type <i>V</i>). Only applicable to files of types <i>L</i> and <i>R</i>.</p>
Sequence field	<p>The descriptor to be used by Natural for logical sequential reading.</p> <p>Determines the sequence in which records are delivered by the READ LOGICAL statement.</p> <p>The GENERATE DDM function will use this field as the default READ LOGICAL field in the Natural data definition module.</p>

ISAM

ISAM Files and Sequential Files (File Types *M* and *S*)

11:46:54

***** P R E D I C T 4.1.1 *****

1999-06-30

- Modify file -

File ID JCA-M

Modified 1998-04-20 at 13:04

Type ISAM file

by JCA

File number 1

in database

Keys ..

Zoom: N

Literal name

Average count

Stability*

Not specified

Data set attributes

External name ..

Zoom: N

Organisation

Size definition

Location

Type*

Unit*

Device

Recfm*

Primary

Volume 1 ..

Reclsize

Secondary

Volume 2 ..

Blksize

Dir blocks

Volume 3 ..

Rounded up N (Y/N)

Volume 4 ..

Contiguous N (Y/N)

Volume 5 ..

Abstract

Zoom: N

EDIT: Owner: N Desc: N Field list: N

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

Parameters

Data Set Attributes	
External name	Name of the physical file in operating system. Up to 250 characters can be specified (using the <i>Zoom</i> option).

Organization Type	The organization of the data set: DA Direct access PO Partitioned PS Sequential <i>blank</i> None of the above applies
Recfm	The record format of the file: F Fixed FB Fixed block FBS Fixed block standard V Variable VB Variable blocked VBS Variable blocked standard U Unblocked <i>blank</i> None of the above applies
Recsize	The record size of the file.
Blksize	The block size of the file.
Rounded up	Y Each space allocation is rounded up to full cylinders.
Contiguous	Y The space allocated to the secondary extent of the file is contiguous with the space allocated to the primary extent.
Size Definition	
Unit	The units in which storage space has been allocated to the file: BL Blocks CY Cylinders TR Tracks
Primary	The number of units of storage space allocated to the primary extent of the file.
Secondary	The number of units of storage space allocated to the secondary extent of the file.
Dir-blocks	The number of blocks reserved for the directory of the file.
Location	
Device	The type of storage device on which the file is located.
Volume 1 – 5	The volume(s) on which the file is located. Up to five volumes can be specified.

Entire System Server

Entire System Server Files and Userviews (File Types *P* and *Q*)

```

15:02:58          ***** P R E D I C T  4.1.1  *****          1999-05-22
                        - Modify File -

File ID ..... PD-P3                      Modified 1998-05-03 at 16:27
Type ..... ENTIRE SYSTEM SERVER                by WRKPR3
File number ..... 1
in database .....
Keys .. AZ-KEYWORD-                          Zoom: N

Literal name .....
Average count
Stability .....*   Not specified

ENTIRE SYSTEM SERVER attributes
  Sequence Field ....*
  Retrieve ..... Y (Y/N)
  Process ..... N (Y/N)

Abstract      Zoom: N
  COPY FROM ACTIVE-JOBS

EDIT:   Owner: N   Desc: N   Field list: N

```

Note:

Parameters not listed below are described in other sections of this manual:

Parameters common to all object types, for example *Keys*, are described under **Global Attributes**, page 6.

Parameters common to all file types, for example *Literal name*, are described under **Common File Attributes**, page 182.

Parameters

Sequence field	The descriptor to be used by Natural for logical sequential reading. Determines the sequence in which records are delivered by the READ LOGICAL statement. The GENERATE DDM function will use this field as the default READ LOGICAL field in the Natural data definition module.
Retrieve	Y Operation system information can be read with this file.
Process	Y Operation system activities can be performed via this file.

Note:

You cannot add files of type *P* with the function *Add a file*. Files of this type are added automatically when Entire System Server is installed.

File-Specific Maintenance

Maintenance functions applying to file objects are called from the *File Maintenance* menu. This menu is called with the command MAINTAIN file or with function code *M* and object code *FI* in a Predict main menu.

The screen is shown on page 179.

Standard maintenance functions applying to files as well as to most other types of Predict Objects are described in Chapter **Maintenance** in the *Predict Reference Manual*. The following file-specific maintenance functions and aspects of standard maintenance functions specific to files are described below:

- Purge file, page 255
- Rename/Renumber file, page 256
- Edit List of fields, page 257
- Force Standard, page 260
- Push Backward, page 260
- Modify Adabas Attributes, page 263
- Modify STARTAB Elements, page 263
- Edit Subquery of a file, page 263

Purge File (Code *P*)

The following files **cannot** be purged with the *Purge File* function.

- all SAG-owned file objects
- Files of type *I* (IMS segment). Files of type *I* can be purged by scratching the IMS database (type *I*) containing the file.

Two lists are displayed before a file is purged:

- A list of objects and generated code which will not be deleted because they are used in some other object which will not be deleted.
- A list of objects and generated code that will be deleted.

The delete operation is then requested. A list of all deleted objects will be displayed after the delete operation has been executed.

DELETE

The following objects are purged if you confirm this function:

- the file and all its userviews
- all fields of the file and its userviews
- generated code of the file and userview
- all links to databases
- all links from the file to children/from parents
- all links from/to objects that are also purged with this function.

In addition,

- all file relations using this file are set to *D* (documented).

When an Adabas C file is purged, all Adabas attributes and STARTAB elements of the file are also deleted.

Note:

A file cannot be deleted if a DDM for the file exists or the file is implemented.

Rename File (Code *N*)

This function is used to change one or several of the following in a single transaction:

- **File ID**
The ID will be changed in all objects that are linked to the file via an association and in all file Relations. Predict checks that the ID of the file is still unique.
- **Logical File number**
Predict checks if all logical file numbers in the database are still unique (except for conceptual databases).
- **File type**
The field list is loaded into the Predict list editor and is checked. It can then be corrected and has to be cataloged. This is especially important if files of type *C* are changed to another type. The following rules apply:
 - If a standard file (File type *Z*) is changed to another file type, all connections to other files are deleted.
 - It is not possible to change the type of a master file if related userviews for this file exist. First connect the userviews to another master file, then change the master file.
- **Master File**
The new related master file can be specified for files of type *J*, *K*, *L*, *Q*, *R*, *U* and *W*.

If a userview is connected to another real file, its field list is loaded into the Predict list editor and is checked. It can then be corrected and has to be cataloged.
- **Logical ADASTAR type**
Only applicable to Adabas C files (File type *A*).
- **Adabas C SQL usage**
Only applicable to Adabas C files (File type *A*).
If set to *Y*, the file is accessible via Adabas SQL Server.

Edit List of Fields (Code L)

The field list editor can be invoked in one of the following ways:

- With *Y* in the field *EDIT Field list* in the EDIT line.
- With the function *Link children* (code *L*) and child type *EL*.
- With the function *Edit list of Fields* (code *E*). This function is not indicated in the *File Maintenance* menu.
- With the command LINK FILE ELEMENT.

Some additional line and editor commands can be used in the list editor:

Line Commands

.E	Skips to the <i>Add</i> or <i>Modify Field</i> screen for the field on the current line.
.E(<i>n</i>)	Skips to the <i>Add</i> or <i>Modify Field</i> screen for the next <i>n</i> fields in the list.

Editor Commands

ADA	Generate two-character field short names for fields that do not already have a short name.
FLIP C	Enables you to enter field IDs with a length of up to 32 characters.
FLIP T	Enables you to enter field IDs and Table/View IDs with a length of up to 32 characters.
FLIP	The default entry fields are displayed.

NU[LL]

Predict automatically sets suppression/null value options for Fields that are added to the dictionary. The value depends on the type of file:

Parameter	File Types		
	All SQL Types * except X	X	Other File Types
Unique option = <i>Unique</i> or Desc. type = <i>Primary</i> or Field format = <i>serial</i>	R	R	N
Others	U	blank	N

Note:

SQL file types include files of type A with parameter *Adabas C SQL usage* set to Y. See list on .

READA

Delete any existing field short names and generate new ones for all fields. This command is only available when editing the field list of a real file or a standard file (not a userview). It is not applicable to field list of SQL files.

SORT ADA

Sort the fields alphabetically by two-character field short name. Fields not on level 1 are not sorted, so group structures are not changed.

SORT

Sort the fields alphabetically by field ID.
Fields not on level 1 are not sorted, so group structures are not changed.

SET ADA [ON]

Apply future SCAN commands to two-character field short names instead of field IDs.

SET ADA OFF

Cancel the above setting.

Note:

All general commands are described under **The Link Editor** of Chapter **Editors in Predict** in the *Predict Reference Manual*.

Comment Lines

When editing field lists of files you can enter comment lines containing descriptive information at any point in the list. The following rules apply:

- Comment lines start with ****** or **/*** in the column *Ty*.
- Comment lines longer than 32 characters are truncated when files are transferred to Natural LightStorm.
- Comment lines are included in generated DDMs if parameter *General comments* of function *Generate DDM* is set to *Y*.
- Comment lines are ignored for all other generation functions.

Force Standard (Code *F*)

This function compares the connected attributes of all fields defined in the specified standard file with the attributes of the connected fields in other files.

If attributes of connected fields are different (and these fields are not marked as non-standard), they are changed to match the standard file if possible. Otherwise, they are marked as non-standard.

Command: FORCE FILE

Push Backward (Code *B*)

This function connects fields in a master file or conceptual file to fields in a standard file. The file must not be a userview or a standard file.

The concepts of this function are described in the section “Rippling”, page 264.

Command: PUSH FILE *<master-file-id>*

```

10:30:50          ***** P R E D I C T  4.1.1  *****          1999-05-10
                  - Push Backward File -

File ID ..... EMPLOYEES

                                Function

                                A   Push back all fields of the file
                                S   Push back selected fields

Function .....

Standard File ..*
Field ID .....          with ADABAS name .. N (Y/N)
with owner ID ...
with keyword ....

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Menu  Canc  S-fi  E-el  M-pr  Print Impl  Last  FLIP  PROF  Next

```

Parameters

File ID	ID of the file to be pushed backward. This value is entered in the <i>File Maintenance Menu</i> and cannot be overwritten here.
Function	A Push back all fields of the file. All fields in the master file are coupled with fields in the standard file. S Push back selected fields. Fields in the master file are displayed for selection. Selected fields are coupled with fields in the standard file.
Standard file	Standard file containing the standard fields to be coupled with the master fields. Use asterisk notation to display a list of standard files for selection.
Field ID	Enter a unique field ID to couple a single field, or display a list of fields for selection by leaving this field blank or using asterisk notation.
with owner ID	The list of master fields for selection can be restricted to fields with the specified owner. Use asterisk notation to specify a range of owners.
with keyword	The list of master fields for selection can be restricted to fields with the specified keyword. Use asterisk notation to specify a range of keywords.
with Adabas name	Y Field attribute <i>Short name</i> is copied from master field to standard field.

Functional Scope

The following rules apply to both options, A and S.

- Fields already connected to a standard field are not processed.
- If a field with the same ID is already present in the standard file but no link exists, a link is established. The field is marked as non-standard if one of the field attributes does not match.
- If a field is not found in the standard file, it is copied to the standard file and a connection is established.

Push Back all Fields of the File

All fields in the master file that meet the selection criteria are coupled to fields in the standard file.

Push Back selected Fields of the File

Fields in the master file that meet the selection criteria are displayed for selection. Selected fields are coupled to fields in the standard file. This is a two-step process.

1. A list of all fields in the master file which meet the selection criteria is displayed. Fields that are not yet coupled to a field in the standard file are marked *will be added* (see screen below).
2. Mark fields to be coupled to fields in the standard file with any non-blank character and press ENTER. Marked fields are coupled immediately and are marked *is connected to ...* in the column *Remarks*.

```
12:52:09          ***** P R E D I C T  4.1.1  *****          1999-05-02
                      - Push backward Field selection -

From File ID .... FILE13
To   File ID .... STANDARD_FILE

M T  L Field name          F  Length  Remarks
-   1 Field1              A    20.0   is connected to FILE12
-   1 Field2              A    30.0   is connected to FILE12
- HY 1 Field3              A    12.0   will be added
```

Modify Adabas Attributes (Code J)

Displays the *Modify Adabas attributes* screen for specifying the physical implementation of an Adabas C file. See page 190 for a description.

Command: MODIFY ADA-ATTR

Modify STARTAB Elements (Code K)

Displays the *Modify STARTAB Elements* screen (see page 198).

Command: MODIFY STARTAB

Edit Subquery of a File (Code Y)

Invokes the expression editor (see page 208). Only applies to SQL views.

Command: EDIT FILE SUBQUERY

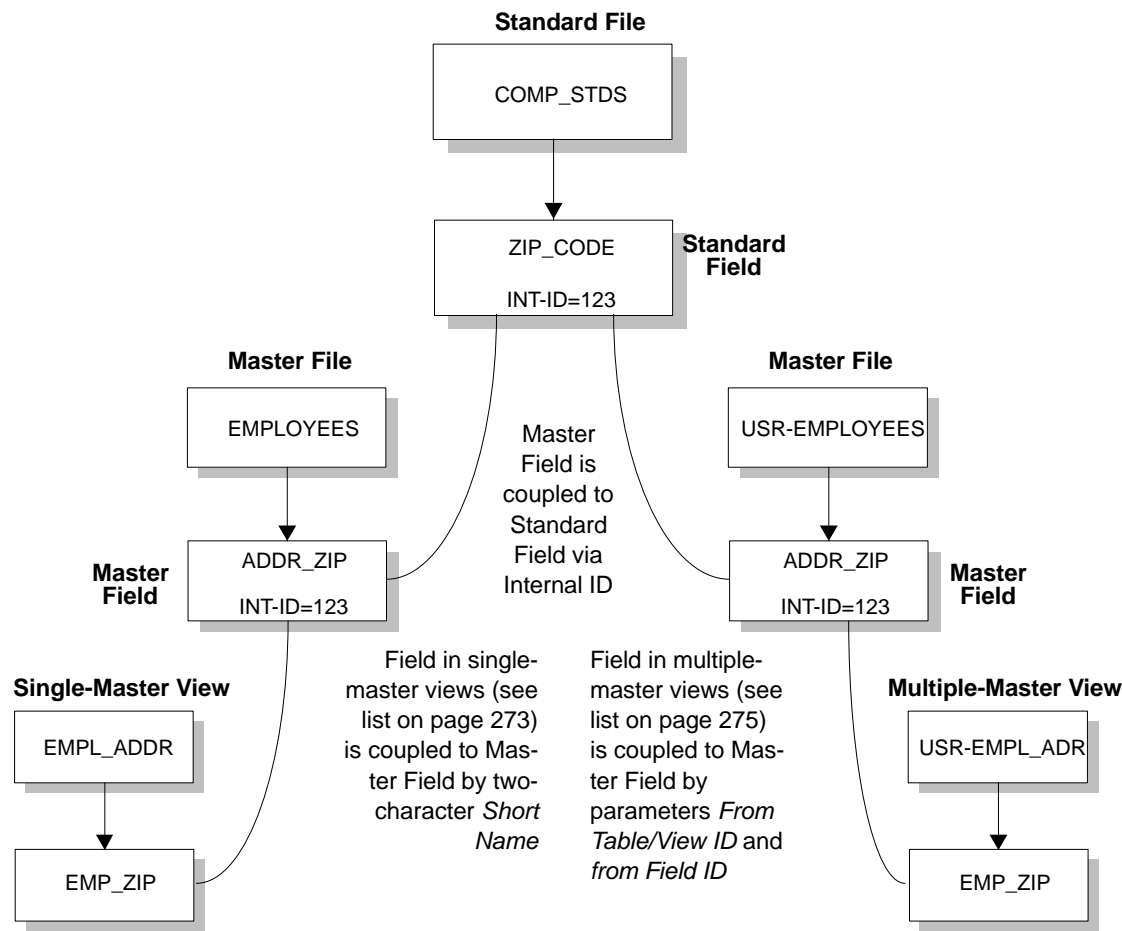
Rippling – Ensuring Consistent Data Definitions

How this Section is Organized

- **Overview** page 265
 - General Recommendations, page 266
 - Listing Rippling Actions, page 266
 - Parameter *Check against standard*, page 267
- **Rippling from Standard Files to Master Files** page 268
 - Creating a Standard File, page 268
 - Coupling of Standard and Master Fields, page 269
 - Functional Scope, page 270
 - Changing Coupled Fields, page 271
 - Uncoupling Standard and Master Fields, page 272
- **Rippling from Master Files to Views/Userviews** page 272
 - Coupling of Master Fields and Fields in Views/Userviews, page 273
 - Functional Scope, page 276

Overview

Predict rippling options can be used to define a standard, hierarchical data structure and to ensure consistent use of this structure throughout an organization: Whenever field definitions on higher levels are changed, all data definitions on lower levels (including views/usersviews) are automatically updated.



General Recommendation

Before you make changes to a standard file, execute the field retrieval function *List Fields related to a Z-file*.

Listing Rippling Actions

Two profile parameters are available for listing rippling actions:

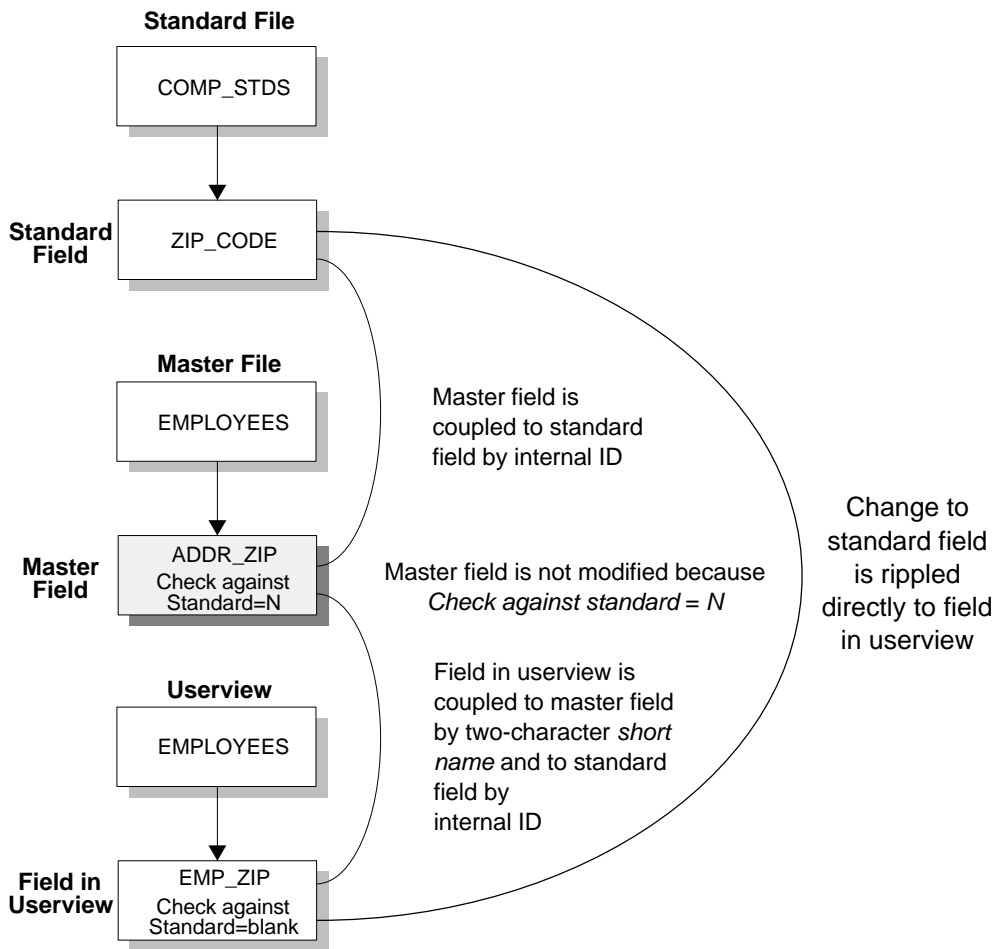
If the parameter *Profile > Maintenance options > List action* is set to *Y*, the modified object is displayed after execution.

If the parameter *Profile > Maintenance options > MORE type-dependent options > List rippling* is set to *Y*, all coupled fields affected by the modification of a higher-level object are listed.

When external objects are generated for the modified file, the external objects are marked as *diff. to documentation*.

Parameter *Check against standard*

This parameter determines whether attribute changes in standard fields are rippled to connected fields. See also page 119.



Rippling from Standard Files

Creating a Standard File

There are two methods of creating a standard file:

- **With Coupling**

Apply the function *Push backward* to a master file. See page 260. The fields in the standard file and in the master file are then coupled. Changes to the standard file automatically result in changes to the master file.

Note:

A field in the master file which is already coupled with a standard field is not copied.

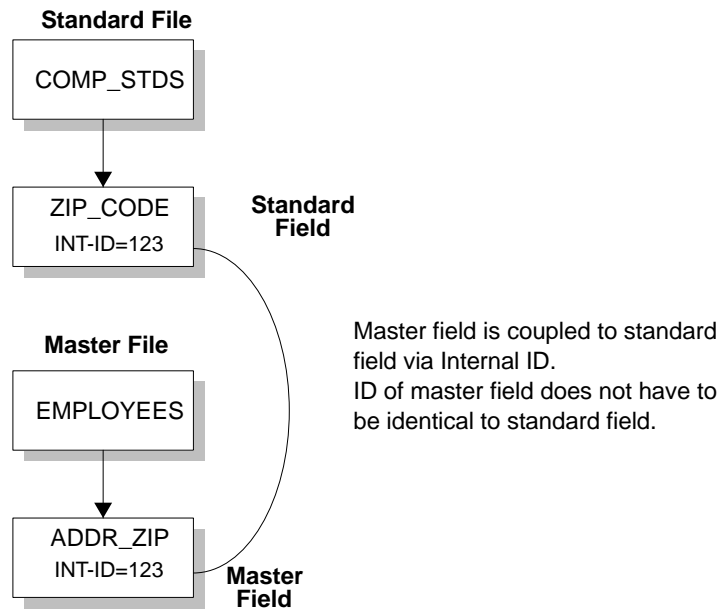
- **Without Coupling**

Create a standard file (file type Z) and copy fields from a master file. Master and standard fields are not coupled and changes to the standard file are not rippled.

Coupling of Standard Fields

Standard fields and connected fields are coupled internally by means of Internal ID.

The coupling remains intact even if the connected field is subsequently renamed.



Functional Scope

The following attributes of a standard field can be rippled to coupled fields at lower levels.

- Field length
- Field format
- Field type
- Suppression option
- Uniqueness option
- Descriptor type (see page 270)

If an attribute is not defined in a standard field (which means the attribute is *blank* if it is alphabetic, or *zero* if it is numeric), no rippling takes place for this attribute and the lower-level object can be modified without restriction. It is therefore possible to have some field attributes defined centrally and others modifiable without restriction at lower levels. See also **Changing Coupled fields**, page 271.

Note:

If one of the attributes above is changed and this change is not compatible with the coupled field, the attribute *Check against standard* of the field is set to *N*. For example: If you change a field type to *HY* (hyperdescriptor), this change is not rippled to coupled fields in DB2 files and the attribute *Check against standard* of the coupled fields is set to *N*.

Rippling the Attribute *Descriptor Type*

The attribute *Descriptor type* of a standard field can have the following values:

- | | |
|--------------|---|
| D | Disallowed. The descriptor type of coupled fields must be <i>blank</i> .
All non-blank descriptor types in coupled fields are set to <i>blank</i> . |
| F | Force. The descriptor type of coupled fields may not be <i>blank</i> .
If a coupled field has a non-blank descriptor type, no rippling is performed.
If a coupled field has descriptor type <i>blank</i> , the descriptor type is set to <i>N</i> and a message is given. |
| <i>blank</i> | Undefined. The descriptor type of coupled fields can be any value, including <i>blank</i> .
No checks are performed, no rippling takes place. |

Rippling Verifications

When the verification list of a standard field is edited, corresponding changes are automatically made in the verification list of every field derived from the standard field. The following rules apply:

- Every verification contained in the verification list of a standard field must also be contained in the verification list of a field coupled to that standard field. However, the sequence of verifications in the lists can differ.
- If a verification is removed from the verification list of a standard field, the verification is automatically removed from the verification lists of all coupled fields.
- If a verification is added to the verification list of a standard field (at any position), the verification is automatically added to the end of the verification list of all coupled fields.
- If the parameter *Check against standard* is set to *N* in connected fields, the checks listed above are not performed.

Changing Coupled Fields

The following rules apply when changing fields at lower levels:

- Attributes not defined in a standard field can be modified in coupled fields.
- Attributes that have been defined in standard fields cannot be modified in coupled fields.
- If an attribute of a coupled field that is defined in the standard field has to be changed, the fields must be uncoupled. See below.

Uncoupling Fields from Standard Fields

Fields can be temporarily or permanently uncoupled from the standard field with the parameter *Check against standard* in the *Modify Field* screen.

- **Temporarily**

Set parameter *Check against standard* to *N*.

The field is uncoupled temporarily from the standard field from which it was derived.

The coupling can be reactivated by resetting *Check against standard* to *blank*.

- **Permanently**

Set the the parameter *Check against standard* to *D*.

The field is uncoupled permanently from the standard field from which it was derived.

The coupling cannot be reactivated with the parameter *Check against standard*. To recouple a field, you must apply the function *Push backward* to the file.

Defining a Standard File as Default File for SELECT Command

With parameter *File for select* in the screen *Profile > Maintenance Options > MORE Type-dependent options* you can specify a default file for the command SELECT. This command can be used in the field List editor of master files or conceptual files.

For single-master views, the default file is the related master file.

Rippling from Master Files to Views/Userviews

The following rules apply:

- Changes to master fields are rippled to fields in userviews that were derived from master files. If the master field is coupled to a standard field, changes to the standard field are rippled to the coupled master field and to the derived field in the userview.
- Changes to fields in userviews are rejected if they are not compatible with the master field.

For example: if a field in a userview is derived from a master field of type *T* (time), the field in the userview can only be changed to format *P* with length 13.

All other changes are rejected.

Coupling of Master Fields and Fields in Views/Userviews

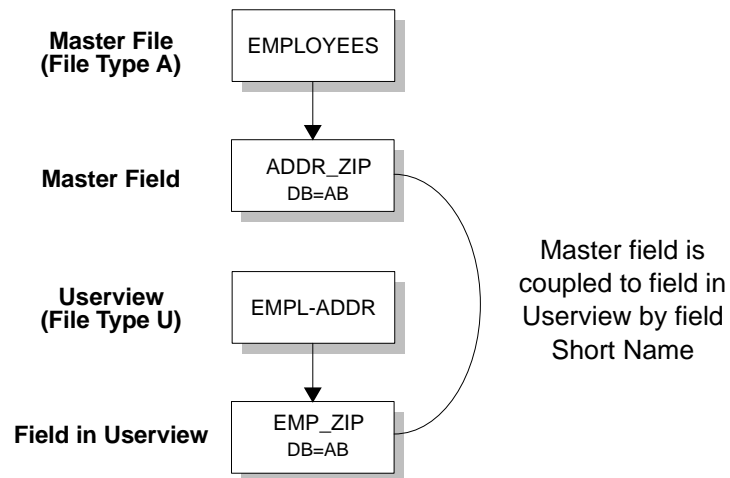
The coupling between master files and views/userviews depends on whether the view is derived from a single master file or from one or several master files.

Single-Master Views

Userviews are derived from one of the following master files:

- Adabas C file
- Physical and logical VSAM files
- IMS Segments
- Entire System Server files

Master fields and fields of Userviews are coupled by field short name (column *DB* in field maintenance screens).



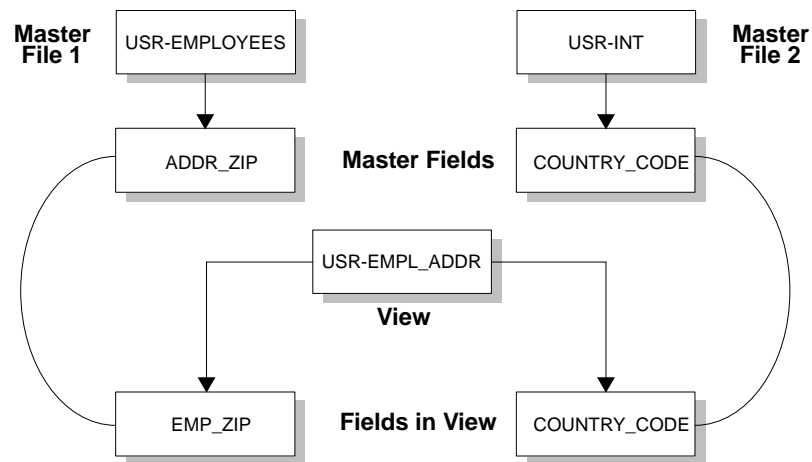
The following table indicates the valid combinations of view types and master file types:

Type of View	Type of Master File
B	A(SQL) AT, B
BV	BT, BV
E	D, E
J	I
JV	JT, JV
K	I
OV	OT, OV
Q	P
R	L
U	A
W	V
XV	XT, XV
YV	YT, YV

Multiple-Master Views

For views which can be derived from several master files, the coupling is established by parameters *from Table/View ID* and *from Field ID* in the field List of the file documenting the view. This applies to the following master file types:

- Adabas C Files (with *SQL usage* set to *Y*)
- Adabas Cluster Tables
- Adabas D Table
- DB2 Table
- INFORMIX Table
- INGRES Table
- ORACLE Table
- SYBASE Table



The coupling above is documented as shown in the field list of file USR-EMPL_ADDR in the screen below.

>		> + Fi: USR-EMP_ADDR		L: 1 S:	
Ty	L	Field ID	from Table/View ID	Field ID	All
* - - - - -					
	1	EMP_ZIP	USR-EMPLOYEES	ADDR_ZIP	
	1	COUNTRY_CODE	USR-INT	COUNTRY_CODE	

Functional Scope

If fields in a master file are modified, views and userviews coupled to these fields are changed accordingly. The following rules apply for this rippling:

Attributes which are always Rippled

The following attributes are always rippled:

- short name (if applicable)
- Field type
- suppression / null value option
- uniqueness option
- character set
- null default option

Attributes which are Rippled if Identical

- The following attributes are rippled if the attribute values in the userview and the master field were identical before the master field was modified:
 - Field ID
 - length, format (both must be identical)
 - max. occ.
 - gr. structur.
 - justify
 - header / edit mask
 - Field/View name name synonym

Abstract

The abstract of a field is rippled according to the setting of the following parameter in the screen *Profile > Maintenance Options > MORE Type-dependent options*:

Ripple abstract	N	Abstract is not rippled.
	T	Abstract is rippled.
	L	Abstract is rippled only if the abstract was identical in the view/userview and the master file before the abstract was changed in the master file.

Rippling Verifications from Master Field to View/Userview

When a verification list of a master field is edited, corresponding changes are automatically made in the verification list of fields in the view/userview derived from the master file. The following rules apply:

- The verification list of a field in a userview does not have to contain all the verifications that are contained in the list of the master file field from which the userview field has been derived.
- If a verification is removed from the verification list of a master field, the verification is automatically removed from the verification list of coupled fields.
- If a verification is added to the verification list of a master field, it is automatically added to the verification list of coupled fields.

File Retrieval

This section is organized as follows:

- The file Retrieval screen, page 279
- File-specific retrieval parameters, page 279
- File-specific retrieval functions
 - Difference of files, page 280
 - List files Related to a file, page 282
- Layout of file Lists, page 283
- Output Options for file Retrieval, page 285

Note:

Standard retrieval functions are described in Chapter **Retrieval** in the *Predict Reference Manual*.

The File Retrieval Screen

The file Retrieval screen below is called with function code *R* and object code FI in a Predict main menu or with command RETRIEVE FILE.

```

15:37:40          ***** P R E D I C T  4.1.1  *****          1999-01-23
Plan  10          - (FI) File Retrieval -                          Profile JCA

Retrieval Type                                Retrieval Type

D  Files
E  Execute retrieval models
C  Dummy/Placeholder files
A  Difference of files

B  Files with parents
O  Files with no parent
T  Files with children
U  Files with no child
R  Files related to a file

Retrieval type ...
Output mode .....* L List

File ID .....
in database .....
External name ...
Restrictions .....*      Profile JCA ,used
Output options ..*      Profile JCA

Files of type .....*
File number .....
Model .....*
Related type .....* EL

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Menu  Canc  S-fi  E-el  M-pr  Print Impl  Last  FLIP  PROF  Next

```

File-specific Retrieval Parameters

in database	Restricts the scope of functions to files and userviews contained in the specified database.
File of type	Restricts the scope of functions to files of the type specified. An asterisk displays a selection window with the valid file types. See page 182 for a list.
File number	Restricts the scope of functions to files with this number.
External name	Name of the physical implementation (DSN, Table names). Can have up to 250 characters, but only the first 50 are evaluated by Predict retrieval functions.

File-specific Retrieval Functions

Difference of Files (Code A)

This function compares files and displays the differences. The file attributes, the fields and the field attributes can be compared. The fields are compared using the field ID.

If a userview is compared with its master file, however, the fields are compared by two-character *Short name*. The userview is always taken as first file, irrespective of which file is entered under *First File ID*.

A screen appears for entering the names of two files and selecting the attributes to be compared.

Command: DIFFERENCE FILE.

```

14:45:50          ***** P R E D I C T  4.1.1  *****          1999-01-23
Plan  10          - Difference of Files -

First File ID ..... JCA-FI1
Second File ID ....

Options
List Fields .....* D Differences only
Compare File attributes ..... N (Y,N)

Mark Field attributes which should be compared.
X the order          X the existence
X abstract           X owner IDs          X keywords
X description        X Field name synonyms X standard File
X verifications      X ADABAS attributes  X Field definition
X NATURAL attributes X specification for 3GL X VSAM attributes

Enter--PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Menu  Canc  S-fi  E-el  M-pr  Print Impl  Last  FLIP  PROF  Next

```

Parameters

First file ID, Second file ID The names of the files to be compared. Asterisk notation can be used to compare one file with many files or two sets of files.

Options

List fields Determines how the result of the comparison operation is to be displayed:
A all fields are listed and differences are marked
D only fields with differences are listed
N a message indicates if differences were found.

Compare file attributes Y File definitions are to be compared.

Field attributes to be compared the order Differences in the order of fields in a file.

Note: The system checks for each field in the list whether the previous field of file 1 is identical to the previous field of file 2. Redefinitions are ignored in the check for previous field. In the example below, EL1 is regarded as previous field of EL2 for both files

FIRST_FILE	SECOND_FILE
-----	-----
Ty L Field ID	Ty L Field ID
-----	-----
1 EL_1	1 EL_1
RE 1 EL_1	RE 1 EL_1
2 EL_11	2 EL_12
2 EL_12	2 EL_11
1 EL_2	1 EL_2

However, the difference in the order of the redefinitions is recognized and the message “Redefinitions are different” is given.

the existence A message is issued if a field exists only in one file.

abstract Abstract of fields.

owner IDs Owners of fields.

keywords Keywords of fields.

Parameters

description	The description of fields.
Field name synonyms	Field-name-synonyms, language-synonym-names.
Standard file	Standard file, non-standard definition.
Verifications	Verifications linked to fields.
Adabas attributes	Security access level, security update level.
Field definition	Descriptor type, level number, field length, field type, max. occurrences, unique option, user exit, EDIT OPTION, SUPPRESSION OPTION, IMS OFFSET, IMS VARIABLE, DB2 field procedure, DB2 field parameter, DB2 master file, DB2 master field, DB2 index cluster, DB2 index subpage, DB2 index bufferpool.
Natural attributes	Edit mask, field headings.
specification for 3GL	Init value, justify, condition names, index name, depend name, structured, VSAM-attributes, alternate index name, VSAM flags.
VSAM attributes	Alternate index name, VSAM flags.

Note:

This command can also be performed in batch mode. See Chapter **Predict Commands** in the *Predict Reference Manual* for a list of keywords and parameters. These keywords are not available online.

Files Related to a File (Code R)

Certain files are considered to be logically related. For example, Adabas C files and userviews; VSAM files and VSAM userviews; logical VSAM files and their userviews. This function displays the following relationships of files:

- master files with their userviews
- userviews with their master files and other userviews of these master files.

For physical VSAM files also the related logical VSAM files are listed, for IMS segments also the IMS segment layouts.

Command: RELATED FILE.

Layout of File Lists

15:46:22	*****	P R E D I C T	4.1.1	*****	1999-12-17
		- List Files -			Page: 1
Cnt	File ID	Type	Fnr	DDM Impl	Other
1	A	S			
2	* A-ADDR-File	A	59	A	
3	* A-ANSP-File	A	84	A	
4	A-File	A	1		
5	A-U-File	U	1		
6	Az-a-File	A	54		
7	AA-TD	D			
8	AA-TS	S			

Meaning of Columns

File ID	ID of the file definition.
Fnr	The physical file number in this database. Only applicable when editing the files of an Adabas C database (type A). If the database is isolated (ADASTAR parameter= <i>I</i>) the physical file number can differ from the logical file number only if the file is an expanded file (ADASTAR type <i>E</i>).
Type	File types and their codes are listed on page 182.
DDM	An asterisk in this column indicates either that a Natural data definition module has been generated for the file or that the file has been used by either Adabas Native SQL or the Predict Preprocessor.
Impl	How a file is implemented: A The file has been loaded into Adabas C C ADACMP definitions have been generated for the file D The file has been implemented in DB2 U UDFs have been generated for the file (IMS) S ADASTAR translation table generated

Meaning of Columns

Other

An asterisk in this column indicates that at least one copy code member for ADASCR, Assembler, C, COBOL, FORTRAN or PL/I or at least one ADAINV card has been generated for the file.

Output Options for File Retrieval

Retrieval Type	D				B				O				T							
													dummies=Y N				dummies=D P			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes	✓				✓				✓				✓				✓			
Adabas sizes																				
Association attributes					✓	✓	✓	✓					✓	✓	✓	✓				
Attributes	✓				✓				✓				✓				✓			
Check expression	✓				✓				✓				✓				✓			
Composed Fields														✓		✓				
Connecting character						✓								✓						
Cover page	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Description	✓				✓	✓			✓				✓	✓			✓			
Display length														✓		✓				
Display modifier	✓				✓				✓				✓				✓			
Dummy/Placeholder														✓		✓		✓		✓
DV-Field expression														✓						
Entry points																				
Extract	✓				✓	✓			✓				✓	✓			✓	✓		
Generation layout														✓		✓				
Adabas C version														✓		✓				
Language														✓		✓				
Alignment/sync.														✓		✓				
Position/Offset														✓		✓				
Counter length														✓		✓				
Compiler														✓		✓				
Replace with syn.														✓		✓				

Retrieval Type	D				B				O				T							
													dummies=Y N				dummies=D P			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓				✓	✓			✓				✓	✓			✓			
Linked Verification														✓						
Mark implementation	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
No. abstract lines	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
Natural options																				
Owner	✓				✓	✓			✓				✓	✓			✓			
With users	✓				✓	✓			✓				✓	✓			✓			
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	✓
Procedure code																				
Rules																				
Show implementation	✓				✓				✓				✓				✓			
Sorted by Field																				
Subquery	✓				✓				✓				✓				✓			
Synonyms														✓		✓				
STARTAB elements	✓				✓				✓				✓				✓			
Trigger	✓				✓				✓				✓				✓			
Use Con-form	✓				✓				✓				✓				✓			
User exit	✓				✓				✓				✓				✓			
3GL specification														✓						

Output Options for File Retrieval (Continued)

Retrieval Type	U				E				C				R			
Output Mode	D		L		T		X		L		D		L			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes	✓															
Adabas sizes																
Association attributes					✓	✓										
Attributes	✓					✓		✓								
Check expression	✓															
Composed Fields																
Connecting character						✓		✓				✓				
Cover page	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Description	✓							✓				✓				
Display length																
Display modifier	✓															
Dummy/Placeholder						✓		✓	✓		✓					
DV-Field expression																
Entry points																
Extract	✓					✓		✓			✓	✓				
Generation layout																
Adabas version																
Language																
Alignment/sync.																
Position/Offset																
Counter length																
Compiler																
Replace with syn.																

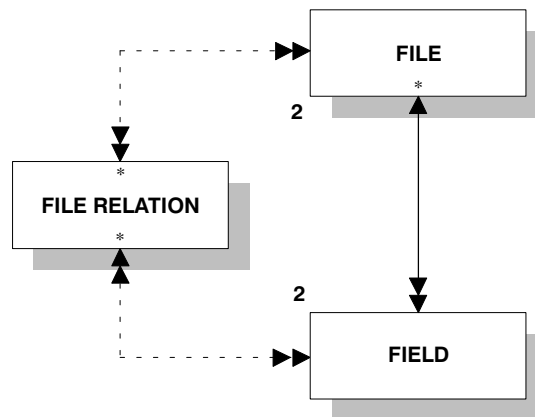
Retrieval Type	U				E				C				R			
Output Mode	D		L		T		X		L		D		L			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓					✓		✓				✓				
Linked Verification																
Mark implementation	✓		✓		✓	✓	✓	✓		✓		✓	✓	✓		
No. abstract lines	✓		✓			✓		✓		✓		✓	✓	✓		
Natural options																
Owner	✓					✓		✓				✓				
With users	✓											✓				
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Procedure code																
Rules																
Show implementation	✓															
Sorted by Field																
Subquery	✓															
Synonyms																
STARTAB elements	✓															
Trigger	✓															
Use Con-form	✓							✓				✓				
User exit	✓															
3GL specification																

FILE RELATION

Note:

This object type was formerly called *relationship*. This name was changed in Predict version 3.3 for reasons of consistency throughout the products Predict, Predict Case and Natural Engineering Workbench (part of the Natural LightStorm package).

The object type *file relation* documents relationships between files. The relationship is established by means of references to fields.



How this Chapter is Organized

- **File Relation Maintenance** page 291
 - The *Add a File Relation* screen, page 292
 - Validity checks for file relations, page 295
- **File Relation Retrieval** page 297
 - File relation-specific retrieval parameters
 - Layout of file relation lists

File Relation Maintenance

The File Relation Maintenance Menu

The *File Relation Maintenance* menu is called with function code *M* and object code *RL* in a Predict main menu or the command MAINTAIN FILE RELATION.

```
13:05:08          ***** P R E D I C T  4.1.1  *****          1999-02-09
Plan    3          - (RL) File relation Maintenance -          Profile JCA

Function                                Function
A  Add a File relation                  D  Display File relation
C  Copy File relation                  L  Link children
M  Modify File relation                 O  Edit owners of a File relation
N  Rename File relation                 S  Select File relation from a list
P  Purge File relation                  W  Edit description of a File relation

Function .....
File relation ID ..
Copy ID .....
for file ID ..... JCA-FI1

Restrictions .....*   Profile JCA ,used           Child type .....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Menu  Canc  S-fi  E-el  M-pr  Print Impl  Last  FLIP  PROF  Next
```

Parameters

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Function	Executes one of the maintenance functions. All standard maintenance functions are described in Chapter Maintenance in the <i>Predict Reference Manual</i> .
for file ID	For the <i>Select</i> function: a file ID can be specified as an additional selection criterion. Asterisk notation is possible.

The Add a File Relation Screen

The screen below is displayed for the *Add a File Relation* function. The *Copy* and *Modify* screens are similar.

13:03:23

***** P R E D I C T 4.1.1 *****

1999-04-25

- Add a File relation -

File relation ... JCA-RL1

Added 1998-11-07 at 15:28

Type* D Documented

by JCA

Keys ..

Zoom: N

Cardinality ..* :

File 1

Minimum ...

File ID* JCA-EL1

Average ...

Field ID ...* JCA-FI1

Maximum ...

File 2

Minimum ...

File ID* JCA-EL1

Average ...

Field ID ...* JCA-FI2

Maximum ...

Constraint attributes

Update type* (none)

Delete type* (none)

Constraint name ..

Usage*

Abstract

Zoom: N

EDIT: Owner: N Desc: N

Parameters

- Note:*
Parameters not listed here are described under **Global Attributes**, page 6.
- File Relation

The ID of the file relation object.
- Type

The type of file relation. Valid values:
C Two files of type A are physically coupled.
D The file relation is only documented.
K Common keys.
This file relation type is only valid for file types YT and YV (SYBASE tables and views).
The field linked to the file relation must have a non-blank descriptor type.

Parameters

	<p>Predict checks whether the number, formats and character sets of the fields – or source fields in the case of superdescriptors – in file 1 and file 2 agree.</p> <p>For SYBASE, you can generate a common key from a file relation of this type.</p> <p>For other database management systems, file relations of this type are used for documentation purposes only.</p> <p>N This file relation type documents the models used by Natural Construct. See Defining File Relations for Objects in Predict in the <i>Natural Construct User's Manual</i>.</p> <p>R Ref. Constraint.</p> <p>Files of type <i>AT, BT, D, JT, OT, X, XT, XV, Y, and YV</i> are connected by referential integrity.</p> <p>S Files of type <i>A</i> are soft coupled.</p> <p>See also table on page 295.</p>
Cardinality	<p>The number of records of each file that is permitted in any occurrence of the file relation. Valid values:</p> <p>1 one (must be one)</p> <p>C none or one (can be one)</p> <p>CM,CN one or one or more (can be many)</p> <p>M, N one or more (must be at least one)</p>
File 1	One of the related files.
File ID, field ID	<p>If the type of file relation is <i>R</i>, the field which is used to link this table must be a primary index (for DB2) or a unique key (for other SQL systems).</p>
File 2	<p>The other related file. If the type of file relation is <i>R</i>, the field which is used to link this table must be one of the following:</p> <ul style="list-style-type: none"> - foreign key (descriptor <i>E</i>) - foreign index (descriptor <i>F</i>) - primary index (descriptor <i>P</i>)
Minimum	<p>The minimum number of occurrences of a field from <i>File 1</i> or <i>File 2</i> in the file relation.</p>

Parameters

Average	The average number of occurrences of a field from <i>File 1</i> or <i>File 2</i> in the file relation.
Maximum	The maximum number of occurrences of a field from <i>File 1</i> or <i>File 2</i> in the file relation.
Constraint attributes	
Update type	<p>The type of constraint to be applied when updating a file relation of type <i>D</i>, <i>N</i> or <i>R</i>.</p> <p>C Cascade</p> <p>R Restricted</p> <p>L Suffix as line number (file relation type <i>D</i> or <i>N</i>)</p> <p>N Renumber suffix (file relation type <i>D</i> or <i>N</i>)</p> <p>S Set NULL.</p>
Delete type	<p>The type of constraint to be applied when deleting a file relation of type <i>D</i>, <i>N</i> or <i>R</i>.</p> <p>C Cascade</p> <p>R Restricted</p> <p>L Suffix as line number (file relation type <i>D</i> or <i>N</i>)</p> <p>N Renumber suffix (file relation type <i>D</i> or <i>N</i>)</p> <p>S Set NULL.</p> <p>D Set default.</p>
Constraint name	The constraint name for a file relation of type <i>D</i> and <i>R</i> .
Usage	<p>Only applicable to file relations of type <i>Natural Construct</i> or <i>Documented</i>.</p> <p>Describes how the file relation is evaluated in Natural Construct:</p> <p>A Construct aggregate.</p> <p>I Construct inheritance.</p>

Validity Checks for File Relations

The validity checks performed by Predict depend on the file relation type:

Code	Type	Applicable for	Validity Checks
C	Physically Coupled	Adabas	May not be any of the following: <ul style="list-style-type: none">- redefined field- group- periodic group- member of a periodic group- hyperdescriptor- phonetic descriptor The first two fields in the file relation must be descriptors with the same length and format.
D	Documented	all types	None
K	Common Keys	SYBASE tables and views	The field linked to the file relation must have a non-blank descriptor type
N	Natural Construct	all types	Both the field and file containing the file relation must be defined in Predict.

Code	Type	Applicable for	Validity Checks
R	Referential Constraint	Adabas Cluster Table, DB2 Table ORACLE Table, Adabas D Table, Informix Table or View	<p>Must be marked in the table of file 1: For <i>file type DB2 table or Informix table/view</i> as primary index (descriptor type <i>P</i>), foreign index (descr. type <i>F</i>) or index (descr. type <i>D</i>), and as unique (unique option <i>U</i>)</p> <p>for <i>file type Adabas cluster table</i> as primary index (descriptor type <i>P</i>);</p> <p>for <i>other file types</i> as unique (unique option <i>U</i>).</p> <p>Must be marked in the table of file 2: For <i>file type Adabas cluster table</i> as foreign index (descr. type <i>F</i>) or foreign key (descr. type <i>E</i>);</p> <p>for <i>other file types</i> as primary index (descr. type <i>P</i>), foreign index (descr. type <i>F</i>) or foreign key (descr. type <i>E</i>).</p>
S	Soft-coupled	Adabas	<p>May not be any of the following:</p> <ul style="list-style-type: none"> - redefined field - group - periodic group - member of a periodic group - hyperdescriptor - phonetic descriptor <p>The first field in the file relation must be a descriptor; the second field must have the same format.</p>

With Predict retrieval functions, file relations between physical files are treated as though they were connected with the userviews of the files.

File Relation Retrieval

File Relation Specific Retrieval Parameter

using file

Restricts the scope of the function to file relations which apply to the specified file. Asterisk notation can be used to specify a range of files.

Layout of File Relation Lists

14:36:22	***** P R E D I C T 4.1.1 *****	1999-02-27
	- List File relation -	

Cnt	File relation ID	Type File 1 File 2
1	AER-TST-SYS1-19	D AER-TST-SYS1 AER-TST-SYS2
2	AER-TST-SYS2-18	D AER-TST-SYS2 AER-TST-SYS1
3	AMMM	D
4	ARH-RL	D ARHTESTCHEN ARH-BT1
5	ARH-RL-FUER-BT-FILE	K ARH-BT1 ARH-BT1
6	ARH-RL-K	K ARH-D1 ARH-D1
7	ARH-RL1	D ARH-123456789012 ARH-123456789012
8	ARH-RL2	R ARH-OT1 ARH-OT1

Meaning of Columns

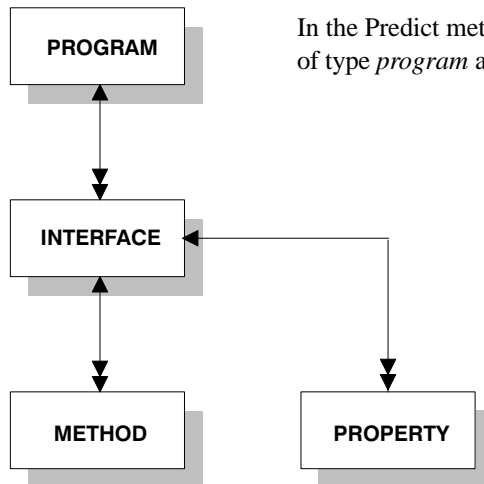
File Relation ID	ID of the file relation object.
Type	The type of file relation. See table on page 295 for list of valid types and codes.
File 1	One of the related files.
File 2	The other related file.

Output Options for File Relation Retrieval

The output options valid for this object type are identical to those for object type *Dataspace*. See page 58.

INTERFACE

This object type, together with object types *method* and *program*, is used to document the Natural programming object class.



In the Predict meta structure, an *interface* can have parents of type *program* and children of type *method* and *property*.

How this Chapter is Organized

- **The Interface Maintenance Menu**, page 299
 - The *Add an Interface* screen, page 300
- **Interface Retrieval**, page 301

The Interface Maintenance Menu

This menu is called with function code *M* and object code *IE* in a Predict main menu, or with the command MAINTAIN INTERFACE.

```
12:33:11          ***** P R E D I C T  4.1.1  *****          1999-10-04
Plan   0          - (IE) Interface Maintenance -          Profile SYSTEM

Function                                Function
A  Add a Interface                      D  Display Interface
C  Copy Interface                      L  Link children
M  Modify Interface                    O  Edit owners of a Interface
N  Rename Interface                    S  Select Interface from list
P  Purge Interface                     W  Edit description

Function .....
Interface ID .....
Copy ID .....

Restrictions .....*   Profile Default,empty   Child type ....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      -      Next  Stop  Last  LnkEl Flip  Print Impl  AdmFi SelFi Prof  Main
```

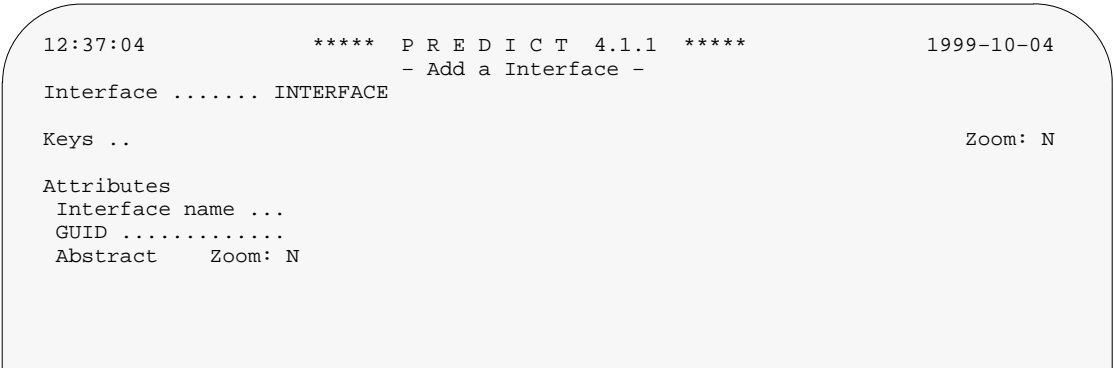
Parameters

The *Interface Maintenance Menu* contains only global attributes. See page 6.

The functions are described in Chapter **Maintenance** in the *Predict Reference Manual*.

The *Add an Interface* Screen

The following screen appears for the function *Add an Interface*. The screens for functions *Copy* and *Modify* are similar.



Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Parameters

Interface	ID of the interface.
Interface name	Name of the interface.
GUID	The globally unique ID of the interface.

Interface Retrieval

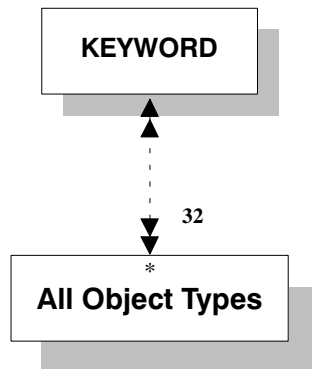
Information on interface objects is gathered using standard retrieval functions. See Chapter **Retrieval** in the *Predict Reference Manual*.

Output Options for Interface Retrieval

The output options valid for this object type are identical to those for object type *dataspace*. See page 58.

KEYWORD

Predict objects of type *keyword* are used to relate objects logically, for example all objects belonging to an application or all objects used in a particular business context.



In the predefined Predict metastructure, a keyword can be related as a child object to objects of all types including other keywords.

How this Chapter is Organized

- **Keyword Maintenance** page 303
 - Keyword Maintenance Menu, page 303
 - The Add/Copy/Modify Keyword Screen
 - Keyword Maintenance Functions
 - Purge Keywords, page 304
 - Link/Unlink Objects, page 305
- **Keyword Retrieval**
 - Keyword Retrieval Functions
 - List Keywords Related to no Object, page 308
 - Cross Reference Keywords, page 308
 - Layout of Keyword Lists, page 308
 - Output Options, page 309

Keyword Maintenance

The *Keyword Maintenance* menu is displayed with function code *M* and object code *KY* in a Predict main menu or the command MAINTAIN KEYWORD.

13:44:12

***** P R E D I C T 4.1.1 *****

1999-03-01

Plan 0

- (KY) Keyword Maintenance -

Profile JCA

Function

Function

A Add a keyword

D Display keyword

C Copy keyword

L Link children

M Modify keyword

O Edit owners of a keyword

N Rename keyword

S Select keyword from a list

P Purge keyword

W Edit description of a keyword

E Link/Unlink objects

Function

Keyword ID

Copy ID

Restrictions ..*

Profile JCA ,used

Child type*

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---

Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next

Parameters

- Note:*
Parameters not listed here are described under **Global Attributes**, page 6.
- Function

Executes one of the functions in the *Keyword Maintenance* menu. Standard maintenance functions are described in Chapter **Maintenance** in the *Predict Reference Manual*. Functions *Purge keyword* and *Link/Unlink objects* are described from page 304.
- Copy ID

For the *Copy* function: the ID of the new keyword.

The Add/Copy/Modify Keyword Screen

The following screen is displayed for the *Add/Copy/Modify Keyword* function.

15:17:45
***** P R E D I C T 4.1.1 *****
1999-11-13

- Add a Keyword -

Key ID JCA-KY2

Keys ..
Zoom: N

Abstract

The parameters are described under **Global Attributes**, page 6.

Keyword Maintenance Functions

Standard maintenance functions are described in Chapter **Maintenance** in the *Predict Reference Manual*. The functions *Purge Keyword* and *Link/Unlink Objects* are described below.

Purge Keyword (Code P)

If you confirm this function with DELETE, the following are deleted:

- the keyword object
- all links to child objects
- all links from parent objects

The number of objects affected by this function is displayed.

Link/Unlink Objects (Code E)

A link between a keyword and a Predict object can be established or deleted directly using the *Link/Unlink objects* function.

Linking or unlinking a keyword and objects is a three-step process:

1. Call the *Link/Unlink objects* screen by entering function code *E* in the *Keyword Maintenance Menu* and specify an object type. Enter an asterisk to display a list of types for selection.
2. Enter search criteria to display a list of objects to be linked or unlinked.
3. Link or unlink objects by entering *L* (link) or *U* (unlink) in the first column.

Steps 2 and 3 are described in more detail below.

Step 2: Specifying Search Criteria

The search criteria depend on the type of object to which a keyword is to be linked. The criteria in the screen below apply when linking databases.

```

15:07:03          ***** P R E D I C T 4.1.1 *****          1999-03-01
Plan    0          - Link/Unlink objects -

Keyword ID ..... JCA-KY1          Added 1998-03-25 at 13:29
                                   by JCA

Link to object type ..* DA ( Database )

Search criteria
Database ID .....
Type .....*
Database number .....
in virtual machine ..

Restrictions .....*   Profile JCA ,used          List option ....* A

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next
  
```

Parameters

Keyword ID	ID of the keyword to be linked.
Link to object type	Type of object to which the keyword is to be linked. Keywords can be linked to any predefined or user-defined object type.
Search criteria	These are object type dependent.
Restrictions	Restrictions can be used to limit the number of objects for selection. See Restrictions in Chapter Retrieval in the <i>Predict Reference Manual</i> .
List Option	The range of objects to be displayed in the list can be restricted as follows: L only objects linked to the keyword are listed U only objects not linked to the keyword are listed A all objects meeting the rest of the criteria are listed (default).

Step 3: Linking or Unlinking Objects

A list of objects which meet the selection criteria is displayed. These objects can be linked or unlinked to the keyword with the following commands in the *CMD* column:

- L link an object
- U unlink an object.

```
15:27:31          ***** P R E D I C T 4.1.1 *****          1999-06-07
          - Link/Unlink objects -

Keyword ID ..... JCA-KY1

CMD L Database                                Type                P-DBnr  ADASTAR Parm.
-   L JCA-DA-A                                ADABAS C              >>> now linked <<<
-   L JCA-DA-D                                DB2
-       JCA-DA-M                                RMS Handler           123   Local
-       JCA-DAX                                DB2
-       JCA-DA1                                ADABAS C              134   Local
-   L JCA-H                                   Gen. SQL Handler      111   Local
```

Objects already linked to the keyword are marked with *L* in the *L* column.

If the parameter *Stay after modify* is set to *Y*, the message >>> *now linked* <<< or >>> *now unlinked* <<< is issued to notify successful execution of the function (as shown above).

If the parameter *Stay after modify* is set to *N*, Predict immediately displays the next page of the selection list (if any) or skips back to the previous *Link/Unlink objects* screen.

Keyword Retrieval

Keyword-specific Retrieval functions

Note:
Standard retrieval functions are described in Chapter **Retrieval** in the *Predict Reference Manual*.

List Keywords Related to no Object (Code Y)

This function lists keywords that are not assigned to any objects.
Command: UNUSED KEYWORD.

Cross Reference Keywords (Code X)

Lists all objects that have specified keywords.
Command: XREF KEYWORD
Valid output mode: Cross reference.

Layout of Keyword Lists

13:32:09	***** P R E D I C T 4.1.1 *****	1999-03-01
	- List Keyword -	

Cnt	Keyword ID	No. of ref
57	CHD-SQLDS	
58	COO	7
59	DATAMODEL-BUSINESS-PARTNER	14
60	DEMO-VERSION	2
61	DEMONSTRATION	
62	DEMONSTRATION2	

Meaning of Columns

No. of Ref. Number of objects to which the keyword is assigned.

Output Options for Keyword Retrieval

Retrieval Type	D				B				O				T							
													<i>dummies=Y N</i>				<i>dummies=D P</i>			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																				
Adabas sizes																				
Association attributes					✓	✓	✓	✓					✓	✓	✓	✓				
Attributes																				
Check expression																				
Composed fields																				
Connecting character						✓														
Cover page	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Description	✓				✓	✓			✓				✓	✓			✓			
Display length																				
Display modifier	✓				✓				✓				✓				✓			
Dummy/Placeholder														✓		✓		✓		✓
DV-field expression																				
Entry points																				
Extract	✓				✓	✓			✓				✓	✓			✓	✓		
Generation layout																				
Adabas version																				
Language																				
Alignment/sync.																				
Position/Offset																				
Counter length																				
Compiler																				
Replace with syn.																				

Retrieval Type	D				B				O				T <i>dummies=Y/N dummies=D/P</i>							
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓				✓	✓			✓				✓	✓			✓			
Linked verification																				
Mark implementation						✓								✓						
No. abstract lines	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
Natural options																				
Owner	✓				✓	✓			✓				✓	✓			✓			
With users	✓				✓	✓			✓				✓	✓			✓			
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Procedure code																				
Rules																				
Show implementation																				
Sorted by field																				
Subquery																				
Synonyms																				
STARTAB elements																				
Trigger																				
Use Con-form	✓				✓	✓			✓				✓	✓			✓			
User exit	✓				✓				✓				✓				✓			
3GL specification																				

Output Options for Keyword Retrieval (Continued)

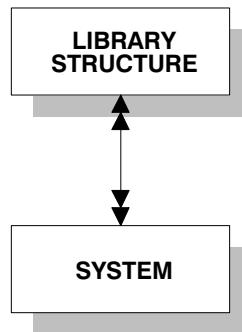
Retrieval Type	U				E				C				Y				X			
Output Mode	D		L		T		X		L		D		D		L		X			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																				
Adabas sizes																				
Association attributes					✓	✓														
Attributes						✓		✓										✓		
Check expression																				
Composed fields																				
Connecting character						✓		✓										✓		
Cover page	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		
Description	✓							✓				✓	✓					✓		
Display length																				
Display modifier	✓												✓							
Dummy/Placeholder						✓		✓	✓		✓									
DV-field expression																				
Entry points																				
Extract	✓					✓		✓			✓	✓	✓					✓		
Generation layout																				
Adabas version																				
Language																				
Alignment/sync.																				
Position/Offset																				
Counter length																				
Compiler																				
Replace with syn.																				

Retrieval Type	U				E				C				Y				X			
Output Mode	D		L		T		X		L		D		D		L		X			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓					✓		✓				✓	✓					✓		
Linked verification																				
Mark implementation						✓		✓		✓		✓						✓		
No. abstract lines	✓		✓			✓		✓		✓		✓	✓		✓			✓		
Natural options																				
Owner	✓					✓		✓				✓	✓					✓		
With users	✓											✓	✓					✓		
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		
Procedure code																				
Rules																				
Show implementation																				
Sorted by field																		✓		
Subquery																				
Synonyms																				
STARTAB elements																				
Trigger																				
Use Con-form	✓							✓				✓	✓					✓		
User exit	✓												✓							
3GL specification																				

LIBRARY STRUCTURE

Programs that are called by another program are not necessarily in the same library as the calling program: it is possible that they are loaded from a steplib at runtime. An object of type *library structure* documents a structure which describes a runtime or development environment (for example libraries for copy code). The corresponding systems are linked as child objects of type *system* to the library structure.

See also Chapter **Steplib Support** in the *Predict Reference Manual* for more information.



In the Predict metastructure, a library structure has the default child type *System*.

How this Chapter is Organized

- **Library Structure Maintenance**
 - The *Library Structure Maintenance* Menu, page 315
 - The *Add/Copy/Modify Library Structure* screen, page 316
 - Function *Link Children*, page 317
- **Library Structure Retrieval** page 317

The Library Structure Maintenance Menu

This menu is called with function code *M* and object code *LS* in a Predict main menu or with command MAINTAIN LIBRARYSTRUCTURE.

```
11:31:50          ***** P R E D I C T  4.1.1  *****          1999-02-22
Plan   3          - (LS) Library structure Maintenance -          Profile JCA

Function                                Function
A  Add a Library structure              D  Display Library structure
C  Copy Library structure               L  Link children
M  Modify Library structure             O  Edit owners of a Library structure
N  Rename Library structure            S  Select Library structure from list
P  Purge Library structure              W  Edit description

Function .....
Library structure ID ..
Copy ID .....

Restrictions .....*   Profile JCA ,used           Child type ....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Menu  Canc  S-fi  E-el  M-pr  Print Impl  Last  FLIP  PROF  Next
```

Parameter

The Library Structure Maintenance menu contains only global attributes. These are described in Chapter **General Information** in the Manual *Predefined Object Types in Predict*, page 6.

These functions are described in Chapter **Maintenance** in the *Predict Reference Manual*. The function *Link children* (with child type *System*) is described in this chapter. See page 317.

The *Add/Copy/Modify Library Structure* Screen

The following screen is called for functions *Add/Copy/Modify Library Structure*:

09:38:53

***** P R E D I C T 4.1.1 *****

1999-06-27

- Add a Library structure -

Library structure JCA-LS1

Keys ..

Zoom: N

Abstract

Zoom: N

EDIT: Owner: N Desc: N

System: N

Parameter

The parameters are described under **Global Attributes**, page 6.

Library Structure Maintenance

Function *Link Children* (Code *L*)

Note:

The following description applies to children of type *system*.

The link list of the library structure contains the main library and the steplibs. The following rules apply:

- The first entry in the link list is the main library, the following entries are steplibs.
- The link list of a library structure can contain up to 10 systems of type *A*.
- The link list can contain additional systems of type *G* (*3GL Application*), but the maximum number of linked systems is 15.
- Dummy objects and systems without an implementation pointer for *Library* are permitted in the link list, but these objects are ignored when the library structure is evaluated for active retrieval function *Program using programs* and all LIST XREF functions.

Library Structure Retrieval

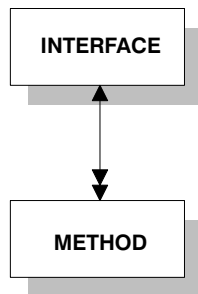
All retrieval functions for library structures are described in Chapter **Retrieval** in the *Predict Reference Manual*.

Output Options for Library Structure Retrieval

The output options available for this object type are identical to those for object type *dataspace*. See page 58.

METHOD

This object type is used to document the methods of an interface.



In the Predict meta structure, a *method* can have parents of type *interface*.

How this Chapter is Organized

- **The Method Maintenance Menu**, page 319
 - The *Add a Method* screen, page 320
- **Method Retrieval**, page 321

The Method Maintenance Menu

This menu is called with function code *M* and object code *MD* in a Predict main menu, or with the command MAINTAIN METHOD.

11:18:41
Plan 0

***** P R E D I C T 4.1.1 *****
- (MD) Method Maintenance -

1999-09-30
Profile SYSTEM

Function

Function

A Add a Method

C Copy Method

M Modify Method

N Rename Method

P Purge Method

D Display Method

L Link children

O Edit owners of a Method

S Select Method from list

W Edit description

Function a

Method ID method

Copy ID

In Interface INTERFACE

Restrictions* Profile Default,empty Child type*

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
- Next Stop Last LnkEl Flip Print Impl AdmFi SelFi Prof Main

Parameters

The *Method Maintenance Menu* contains only global attributes. See page 6.

The functions are described in Chapter **Maintenance** in the *Predict Reference Manual*.

The *Add a Method* Screen

The following screen appears for the function *Add a Method*. The screens for functions *Copy* and *Modify* are similar.

11:21:30

***** P R E D I C T 4.1.1 *****

1999-09-30

- Add a Method -

Method METHOD

in Interface ...* INTERFACE

Keys ..

Zoom: N

Attributes

Method name

Abstract Zoom: N

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Parameters

Method	ID of the method.
Method name	Name of the method.

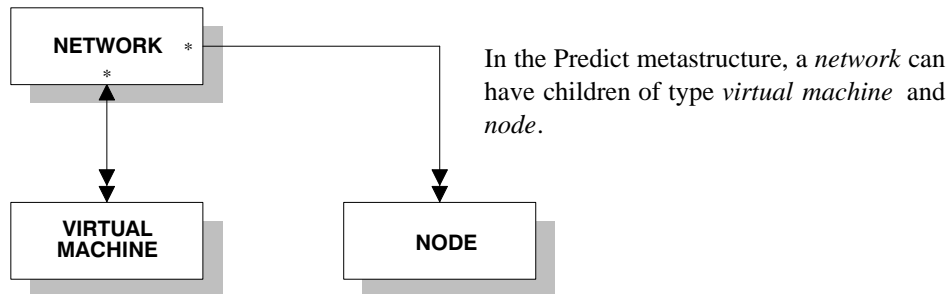
Method Retrieval

Information on method objects is gathered using standard retrieval functions. See Chapter **Retrieval** in the *Predict Reference Manual*.

Output Options for Method Retrieval

The output options valid for this object type are identical to those for object type *dataspace*. See page 58.

NETWORK



The location of a database must be specified by linking each database to an object of type *virtual machine* and each virtual machine to an object of type *network*. A *current network* can be defined in the *Miscellaneous* section of the *General defaults* of Predict or with the command `SET VM <virtual machine ID>`.

The current network will be taken as default for virtual machine objects if no network is specified.

Links between virtual machines and networks are established by entering the network in the parameter *in network* of the virtual machine. A virtual machine cannot be linked to a network using the link editor.

See **Defining the Distribution of Data in Predict** in Chapter **Adabas Star** in the Manual *Predict and Other Systems* for a description of how to define the distribution of data.

How this Chapter is Organized

- **Network Maintenance**
 - The Network Maintenance Menu, page 323
 - The *Add a Network* Screen, page 324
 - Function *Purge Network*, page 324
- **Network-Specific Retrieval**, page 325
 - ADASTAR Numbers, page 325
 - Layout of Network Lists, page 326

The Network Maintenance Menu

The *Network Maintenance* menu is called with function code *M* and object code *NW* in a Predict main menu, or with the command MAINTAIN NETWORK.

```
15:13:21          ***** P R E D I C T   4.1.1   *****          1999-07-22
Plan    3          - (NW) Network Maintenance -          Profile SYSTEM

Function                                     Function
A  Add a Network                             D  Display Network
C  Copy Network                              L  Link children
M  Modify Network                            O  Edit owners of a Network
N  Rename Network                           S  Select Network from list
P  Purge Network                             W  Edit description

Function .....

Network ID .....
Copy ID .....

Restrictions .....*   Profile JPE ,empty           Child type ....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Next Canc S-fi E-el M-pr Print Impl Let  FLIP PROF Menu
```

Parameters

The parameters are described under **Global Attributes**, page 6.

All functions are described in Chapter **Maintenance** in the *Predict Reference Manual*. The function *Purge Network* is described on page 324.

The Add a Network Screen

The following screen is displayed for the *Add a Network* function. The *Copy* and *Modify* screens are similar.

```

16:06:57          ***** P R E D I C T 4.1.1 *****          1999-11-13
                        - Add a Network -
Network ..... JCA-NW22
Keys ..                                     Zoom: N

Abstract      Zoom: N
  
```

The parameters are described under **Global Attributes**, page 6.

Network-Specific Maintenance

Standard maintenance functions are used for maintaining networks. These are described in Chapter **Maintenance** in the *Predict Reference Manual*.

The special rules applying to function *Purge Network* are described below.

Purge Network (Code *P*)

The following rules apply:

- A network that is linked to a virtual machine cannot be deleted.
- The network defined as *current network* in the *General defaults* cannot be deleted.

Network Retrieval

Standard retrieval functions are described in Chapter **Retrieval** in the *Predict Reference Manual*. The network-specific function *ADASTAR Numbers* is described below.

ADASTAR Numbers (Code N)

Displays information on the use of ADASTAR numbers in list form (see sample output below).

Parameters

Network ID	Restricts the report to ADASTAR numbers used in the given network.
Start value / End value ADASTAR number	Restricts the report to ADASTAR numbers in the given range. A window appears for specifying <i>Start value</i> and <i>End value</i> . The ADASTAR number can be specified directly or by the logical DBnr and Fnr from which it is calculated. See Chapter Adabas Star in the Manual <i>Predict and Other Systems</i> for a description of how the ADASTAR number is calculated.

Sample Output

```
09:53:36          ***** P R E D I C T  4.1.1  *****          1999-11-23
                        - List ADASTAR Numbers -                               Page:   4

Network ID ..... BOE-NW

Cnt  Object Id                                Object type  ADASTAR
                                Lnr L-DBnr L-Fnr User
1 BOE-DA-ISO-2                        Database      256      1      0
  Type ... ADABAS,Isolated              ..... Thru .....
  reserve ADASTAR number                 511      1    255
2 BOE-FI03                             STARTAB e1    513      2      1
3 HEB-EDT                             STARTAB e1    515      2      3
4 BOE-FI07                             Phys. file   517      2      5
5 BOE-FI-E-02                         Phys. file   518      2      6
```

Meaning of Columns

Object ID	<p>ID of object referencing the ADASTAR number. The following information on the object may be displayed.</p> <p><i>Type ...</i></p> <p>For databases: the <i>ADASTAR parameter</i> of the database (<i>isolated, local, translator, no translator</i>).</p> <p><i>reserve ADASTAR number</i></p> <p>For databases: a range of ADASTAR numbers is reserved depending on the <i>L-DBnr</i>. The range is calculated as follows:</p> $256 * L-DBnr \leq ADASTAR \text{ number} \leq 256 * L-DBnr + 255$ <p>For example: if <i>L-DBnr</i> is 2, the range 512 – 767 is reserved for ADASTAR numbers.</p>
Object type	<p>Type of object referencing the ADASTAR number. Can be one of the following:</p> <p><i>Database</i></p> <p><i>Phys. file</i></p> <p><i>STARTAB el</i></p>
Lnr	<p>ADASTAR number calculated from <i>L-DBnr</i> and <i>L-Fnr</i> (see page 61).</p>
L-DBnr / L-Fnr	<p>Logical database and file number identifying the file uniquely in a network.</p>
User	<p>User specified with the <i>STARTAB</i> element to restrict access to the data.</p>

Layout of Network Lists

Network lists contain the network IDs.

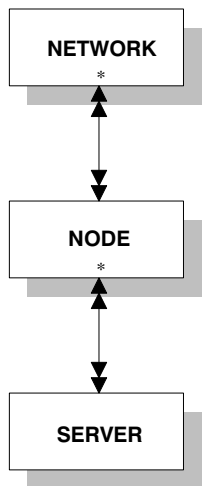
Output Options for Network Retrieval

The output options valid for this object type are identical to those for object type *dataspace*. See page 58.

NODE

This object type, together with object type *server*, is used to document remote procedure calls.

An object of type *node* documents the physical machine containing the server.



In the Predict metastructure, a *node* can have parents of type *network* and children of type *server*.

How this Chapter is Organized

- **The Node Maintenance Menu** page 329
 - The *Add a Node* screen, page 330
- **Node Retrieval** page 331

The Node Maintenance Menu

This menu is called with function code *M* and object code *NO* in a Predict main menu, or with the command MAINTAIN NODE.

12:33:11
Plan 10

***** P R E D I C T 4.1.1 *****
- (NO) Node Maintenance -

1999-10-04
Profile JCA

Function

A Add a Node
C Copy Node
M Modify Node
N Rename Node
P Purge Node

Function

D Display Node
L Link children
O Edit owners of a Node
S Select Node from list
W Edit description

Function

Node ID

Copy ID

In Network

Restrictions* Profile JCA ,used Child type*

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next

Parameters

The *Node Maintenance Menu* contains only global attributes. See page 6.

The functions are described in Chapter **Maintenance** in the *Predict Reference Manual*.

The *Add a Node* Screen

The following screen appears for the function *Add a Node*. The screens for functions *Copy* and *Modify* are similar.

12:37:04

***** P R E D I C T 4.1.1 *****

1999-10-04

- Add a Node -

Node JCA-NO1

in Network*

Keys ..

Zoom: N

Node name

Abstract Zoom: N

EDIT: Owner: N Desc: N

Server: N

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Parameters

Node ID	ID of the node.
in Network	ID of the parent network.
Node name	Name of the node. Up to 8 characters.
EDIT: Server.	Y Edit the server list. An asterisk in front of this field indicates that a server list for this node exists.

Node Retrieval

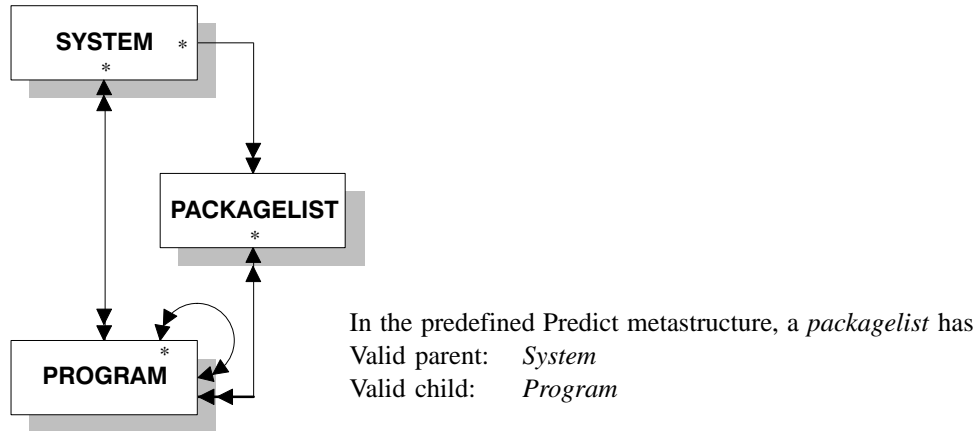
Information on node objects is gathered using standard retrieval functions. See Chapter **Retrieval** in the *Predict Reference Manual*.

Output Options for Node Retrieval

The output options valid for this object type are identical to those for object type *dataspace*. See page 58.

PACKAGELIST

The Predict object type *packagelist* is used to document DB2 packages.



Note:

Packagelists of type *T* and packagelists of type *S* are related using the parameters *Collection name* and *Location name*.

How this Chapter is Organized

- **Packagelist Maintenance**
 - The Packagelist Maintenance Menu, page 333
 - The *Add a Packagelist* Screen, page 335
 - Function *Purge Packagelist*, page 337
- **Packagelist Retrieval**, page 338
 - Packagelist-specific retrieval parameter, page 338
 - Layout of packagelist lists, page 338

The Packagelist Maintenance Menu

The following screen is displayed with function code *M* and object code *PG* in a Predict main menu or the command MAINTAIN PACKAGELIST.

```
11:47:47          ***** P R E D I C T  4.1.1  *****          1999-03-02
Plan    0          - (PG) Packagelist Maintenance -          Profile JCA

Function                                Function
A  Add a Packagelist                    D  Display Packagelist
C  Copy Packagelist                     L  Link children
M  Modify Packagelist                    O  Edit owners of a Packagelist
N  Rename Packagelist                    S  Select Packagelist from a list
P  Purge Packagelist                     W  Edit description of a Packagelist

Function .....

Packagelist ID ...                      Packagelist type ....*
Copy ID .....
in system .....

Restrictions ....*   Profile JCA ,used   Child type .....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next
```

Parameters

Function	Executes one of the maintenance functions. Standard maintenance functions are described in Chapter Maintenance in the <i>Predict Reference Manual</i> . The function <i>Purge</i> is described on page 337.
Packagelist ID	Identifier of the Predict packagelist object. See Naming Conventions on page 6.

Parameters

Packagelist type	<p>Type of packagelist. Valid values:</p> <p>T Total collection. Packagelists of type <i>T</i> provide an over-view of all packages used in a collection. The parameters <i>Collection name</i> and <i>Location name</i> are mandatory for packagelists of type <i>T</i>.</p> <p>Q Database request module (DBRM). Packagelists of type <i>Q</i> contain one DBRM which is directly bound to the plan.</p> <p>S Subcollection. Packagelists of type <i>T</i> and packagelists of type <i>S</i> are connected using the parameters <i>Collection name</i> and <i>Location name</i>. Each package in a packagelist of type <i>S</i> is also contained in a packagelist of type <i>T</i>.</p>
Copy ID	<p>For <i>Copy</i> function: ID of the packagelist to be created.</p>
in system	<p>In DB2, packagelists are used by application plans. Applications plans are documented in Predict with objects of type <i>system</i>, subtype <i>P</i>. Hence the attribute <i>in system</i> is used to document by which plan a packagelist is used.</p>
Child type	<p>For function <i>Link children</i>: Objects of this type are to be linked to the packagelist. Valid values: <i>file</i>, <i>program</i> and <i>user-defined</i>.</p>

The *Add a Packagelist* Screen

The screen is displayed for the *Add a Packagelist* function. The *Copy* and *Modify* screens are similar.

```

09:45:26          ***** P R E D I C T  4.1.1  *****          1999-03-03
                        - Add a Packagelist -

Packagelist ID .. JCA-PG2
Type .....* T
in system .....* BOE-SY-P
Keys ..                                           Zoom: N

Packagelist attributes
  Collection name ...
  Location name .....

Abstract      Zoom: N

```

Predict ensures the consistency of related packagelists (types *T* and *S*):

- If a package is purged from a packagelist of type *T*, it is purged automatically from corresponding packagelists of type *S*.
- If a package is added to a packagelist of type *S*, it is added automatically to the corresponding packagelist of type *T*.

Parameters

Parameters not listed here are described with the *Packagelist Maintenance* menu on page 333.

Packagelist attributes

Collection name

From version 2.3 of DB2 and above, packages are always referenced via *collections*.

A collection is a virtual summary of packages, used to simplify references to packages. In Predict, collections are documented as attributes of packagelists. Packagelists are grouped by including several packages to the same collection.

A collection is documented in Predict with the attributes *collection name* and *location name*. A collection name can be up to 18 characters long.

Parameters

Location name	Together with collection names, location names identify collections uniquely. A location name can be up to 16 characters long.
EDIT Program	<p>Y Edit Program list of the packagelist. Programs of the following types can be linked to packagelists:</p> <ul style="list-style-type: none"> - Program (type <i>P</i>) - Subprogram (type <i>N</i>) - Function (type <i>F</i>) <p>The Predict Link Editor is invoked. See Chapter Editors in Predict in the <i>Predict Reference Manual</i>.</p>

Packagelist Specific Maintenance

Purge Packagelist (Code *P*)

The following rules apply:

- If you confirm this function with DELETE, the following objects are deleted:
 - the packagelist object
 - all links to child objects
 - all links from parent objects
- With packagelists of type *T*, all packagelists of type *S* connected to the packagelist via the attributes *Collection/Location name* are deleted as well. You must enter an additional confirmation before deleting these additional objects.

Packagelist Retrieval

Packagelist-specific Retrieval Parameter

in system

System to which the packagelist is linked.

Layout of Packagelist Lists

```

09:50:10          ***** P R E D I C T 4.1.1 *****          1999-03-03
                        - List Packagelist -

-----
Cnt  Packagelist ID          T Collection          Location

1  AMA-PG1                  T DSDS                  ERE
2  AMMM                    T CVXCV                  XCVXC
3  ARH-PA-1                 T COLL                  LOC
4  ARH-PA-2                 T COL                   LOC
5  BA-PG                    T JKJ                   KJKKK

```

Meaning of Columns

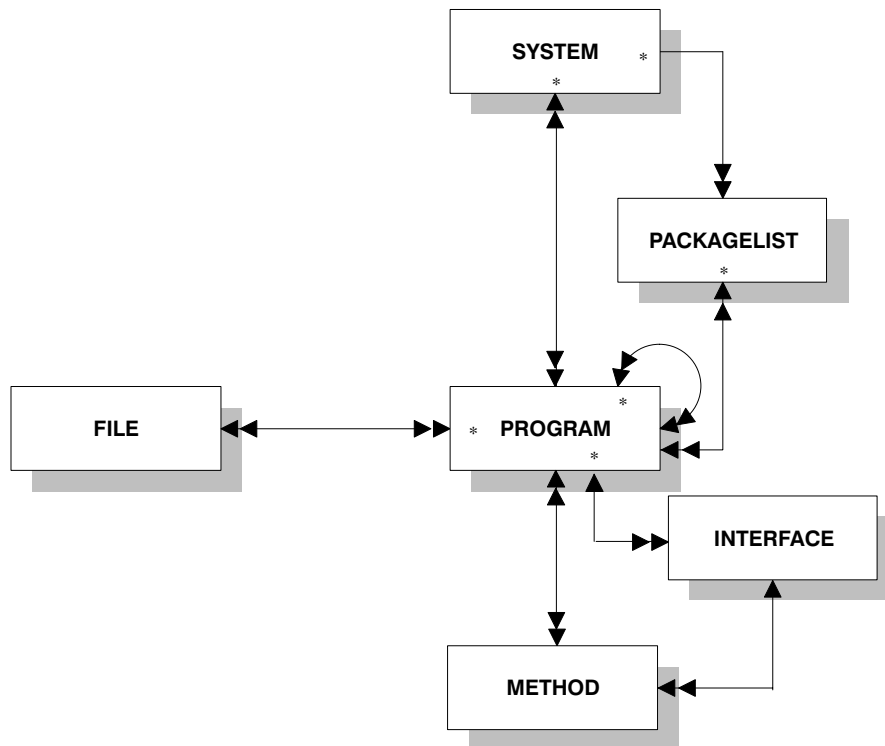
T	Type of packagelist: Q DBRM T Total collection S Subcollection
Collection	Collection of the packagelist. Packagelists of type <i>T</i> and of type <i>S</i> that belong together have the same collection and location name.
Location	Location of the packagelist. Packagelists of type <i>T</i> and of type <i>S</i> that belong together have the same location and collection name.

Output Options for Packagelist Retrieval

The output options valid for this object type are identical to those for object type *dataspace*. See page 58.

PROGRAM

Predict knows more than a dozen different types of programs, ranging from *parameter data area* to *Natural Expert model*. About a dozen different programming languages are supported.



In the predefined Predict metastructure, a *program* can have parents and children of the following types:

Valid parents:

Packagelist, program, system (default parent)

Valid children:

File (first default child), *program* (second default child), *interface* and *method*.

How this Chapter is Organized

- **Program Maintenance**
 - The Program Maintenance Menu page 342
 - The *Add a Program* Screen, page 346
 - System Programs, page 351
 - Programs of Type *Dynamic*, page 351
 - Editing Child Lists, page 352
 - Generating Database Request Modules (DBRMs) from Objects of Language Q (Static SQL), page 356
 - Program-specific Maintenance Functions
 - Function *Purge Program*, page 357
 - Function *Redocument Program*, page 357
 - Edit procedure code of a program, page 365
- **Program Retrieval** page 366
 - Program-specific Retrieval Parameters, page 366
 - Layout of Program Lists, page 367
 - Valid Output Options, page 368

The Program Maintenance Menu

The *Program Maintenance* menu is with function code *M* and object code *PR* in a Predict main menu or with the command MAINTAIN PROGRAM.

```

14:16:56          ***** P R E D I C T  4.1.1  *****          1999-02-21
Plan   3          - (PR) Program Maintenance -          Profile JCA

Function                                Function
A  Add a Program                        D  Display Program
C  Copy Program                        L  Link children
M  Modify Program                      O  Edit owner of a Program
N  Rename Program                     R  Edit entry-points
P  Purge Program                      S  Select Program from a list
W  Edit description of a Program       X  Redocument Program
                                      Y  Edit procedure code of a Program

Function .....
Program ID .....
Copy ID .....
in system .....
Member .....
Library .....
Restrictions ....*   Profile JCA ,used

Function .....
Program of type ....*
Language .....*
User system Fnr .....
User system DBnr ....
Child type .....*

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next

```

Parameters

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Function	Executes one of the maintenance functions. Standard maintenance functions are described in Chapter Maintenance in the <i>Predict Reference Manual</i> . Program-specific functions are described below on page 352.
----------	---

Parameters

Program of type

For the *Select* function:

Program type can be specified as an additional selection criterion.

For the *Add* and *Copy* function:

Program type of the new object. Value is passed to the *Add/Copy Program* screen. Valid values:

A	Parameter data area
C	Copy Code
D	Documented program
E	External program
F	Function
G	Global data area
H	Help routine
I	Dynamic (see page 351)
J	Job
L	Local data area
M	Map/Help map
N	Subprogram
O	Natural command processor
P	Main Program
R	SQL Procedure
S	Natural subroutine
T	Dialog
Y	Expert model
4	Class
5	Resource
<i>blank</i>	Undefined

Language

For the *Select* function:

language can be specified as an additional selection criterion.

Parameters

For the *Add* and *Copy* function:
 language of the new object. Value is passed to the *Add/Copy*
Program screen. Valid values:

B	BAL (Assembler)
C	COBOL
E	Natural EL
F	FORTRAN
G	ADA
H	C
J	Job Control Language
N	Natural
O	Other
P	PL/I
Q	Static SQL
S	SQL Procedure Language
Z	System Program, see page 351
0	Language 0
1	Language 1
2	Language 2
3	Language 3
<i>blank</i>	Unknown

Member, Library, User system Fnr / DBnr

For the *Select* function: implementation pointer values can be
 used to restrict the scope of objects to be processed. Only those
 Predict program objects will be processed that document
 members meeting the specified *Member/Library/Fnr/DBnr*
 parameters.

Member	Member documented by the Predict program.
--------	---

Library in which the member is stored. Either a Natural library or one of the following can be specified:

SYSADA	for ADA
SYSBAL	for ASSEMBLER
SYSCCC	for C
SYSCOB	for COBOL
SYSFOR	for FORTRAN
SYSPLI	for PL/I
SYSSTA	for Static SQL
SYSSYS	for system programs
user-defined	library of a 3GL application environment; must be documented in an object of type <i>System</i>

Number of the user system file.

Number of the database in which the user system file is implemented.

The *Add a Program* Screen

The screen is displayed for the *Add a Program* function. The *Copy* and *Modify* screens are similar.

14:21:11

***** P R E D I C T 4.1.1 *****

1999-02-21

- Add a Program -

Program ID JCA-PR-NEW

Type* P Program

in system*

Keys ..

Zoom: N

Program attributes

Language* All

Mode* (none)

Load-Lib

Implementation pointer

Member User system Fnr ...

Library User system DBnr ..

NAT-Func

('*' to get NAT-Function name from Xref data)

Procedure name ..

Abstract

Zoom: N

Authors

Zoom: N

EDIT: Owner: N Desc: N Progr.: N Files: N MORE: * Attributes: Y

Parameters

Note:

Parameters not listed here are described under **Global Attributes**, page 6.

Program ID	The ID of the program object.
Type	Program type. Must suit the language. The language can be left blank (undefined) for any program type. Enter an asterisk for list of possible values. See 353 for a table of valid program type/language combinations.
in system	ID of the system to which the program is linked. If the program is connected to more than one system, >>>multiple<<< is displayed in this field in the <i>Modify Program</i> function and the field is protected.

Parameters

Program attributes

Language	The language in which the program is written. Enter an asterisk for list of possible values. See 353 for a table of valid program type/language combinations.
Mode	Mode of operation in which the program is used. A All (both online and batch modes) B Batch mode O Online <i>blank</i> Undefined

Load-lib	The load library.
----------	-------------------

Implementation Pointer

Member	Member documented by the Predict program (not applicable to programs of type 5).
Library	The name of the library in which the member is stored (not applicable to programs of type <i>D</i>). - For Natural programs: see the table on page 353. - For 3GL programs: - one of the standard 3GL libraries (see description of the <i>Library</i> parameter, page 345) - any library, provided that it is documented in a Predict system object of type <i>G</i> .

User system Fnr	The number of the user system file. For 3GL programs, the number is always 255.
-----------------	--

User system DBnr	The number of the database in which the user system file is located. For 3GL programs, the number is always 255.
------------------	--

NAT-Func	Applicable only to Natural subroutines (type <i>S</i>). The name of the function of the subroutine (DEFINE SUBROUTINE). If an asterisk is entered, Predict derives the function name from Xref data if Xref data exists for the specified member.
----------	--

Procedure name	Only for programs of type <i>R</i> and Language <i>S</i> . This name must comply with SQL naming conventions. See Chapter Adabas D and Other SQL Systems in the Manual <i>Predict and Other Systems</i> .
----------------	---

Defining More Attributes of Programs

If MORE Attributes is set to “Y”, a window is displayed which contains the following additional attributes for selection:

Entry points	Entry points are to be modified. This is valid only for programs written in certain languages. See Chapter Editors in Predict in the <i>Predict Reference Manual</i> .
SQL procedure code	Only for programs of type <i>R</i> and Language <i>S</i> . The SQL Procedure Editor is called.
Class definition	see page 349
Resource definition	see page 350

The following rules apply:

- Only those types of additional attributes appear in the window that apply to the type of program. For example: the option *Class definition* is not contained in the list when a program of type Resource is processed.
- More than one choice can be made at a time. The respective input maps are then displayed one after the other.

The additional attributes are described in the following sections.

Programs of Type Class

```
DIC1117 PROGRAM NOT MODIFIED.
14:33:43          ***** P R E D I C T 4.1.1 *****          1999-09-13
                        - Modify Program -
Program ID ..... HEB-CLASS          Modified 1999-08-06 at 13:31
                                   by HEB

Class definition
Name .....
GUID .....
Version .....

EDIT:   Owner: N   Desc: N   Progr.: N   Files: N
```

Parameters

Class definition	
Name	The name of the class.
GUID	The globally unique ID of the class.
Version	The version number of the class.

Programs of Type Resource

DIC1117 PROGRAM NOT MODIFIED.
 14:35:26 ***** P R E D I C T 4.1.1 ***** 1999-09-13
 - Modify Program -
 Program ID HEB-RES Modified 1999-09-13 at 08:23
 by HEB

 Resource definition
 File name fi-na

 Library
 User system Fnr ..
 User system DBnr .

 EDIT: Owner: N Desc: N Progr.: N Files: N

Parameters

Resource definition	
File name	File name documented by the Predict program.
Library	The name of the library in which the file name is stored.
User System Fnr	The number of the user system file.
User System DBnr	The number of the database in which the user system file is located.

The type of Resource can be documented in the language field of a Predict program object. There is a user exit program U-PGMLAN that allows dynamic extension of possible languages in each installation.

System Programs

Programs that are only available as object code and hence have no language are documented with programs of type *E* (external object) and language *Z* (system program). Predict creates Xref data for these so called system programs because neither the preprocessor nor Natural can create Xref data for object code.

The implementation pointer for a *system program* has to be specified explicitly. One entry point (with the ID of the program object) is created by Predict, additional entry points have to be specified manually.

Programs of Type *dynamic*

Programs of type *dynamic* are used to document calls of programs of the same name from different steplibs depending on the library structure. The following rules apply:

- Because programs of type *dynamic* document any number of implemented members, no check is performed as to whether the members documented by the program are actually implemented.
- With the active retrieval function *Programs using programs*, programs of type *dynamic* are ignored as current objects.
- Programs of this type can only have children of type *program*.

Program Maintenance

Standard maintenance functions are described in Chapter **Maintenance** in the *Predict Reference Manual*. The following functions are described below:

- Editing child lists
- Generating Database Request Modules (DBRMs) from objects of language Q (Static SQL), page 356
- Function *Redocument Program*, page 357
- Function *Edit procedure code of a program*, page 365.

Editing Child Lists

To edit the lists of entry points, programs and files linked to a program, call the object list editor using one of the following methods:

- Enter *Y* in the field *EDIT Entries/Progr./Files* at the bottom of the *Add/Copy/Modify program* screens. An asterisk before one of these fields indicates that the program already contains a list of entry points, programs or files.
- Call the function *Edit entry points* or *Link children* in the *Program Maintenance Menu* (codes *L* and *R*).
- Enter command EDIT PROGRAM ENTRY <program ID> or LINK CHILDREN.

Overview of Language-Specific Program Types

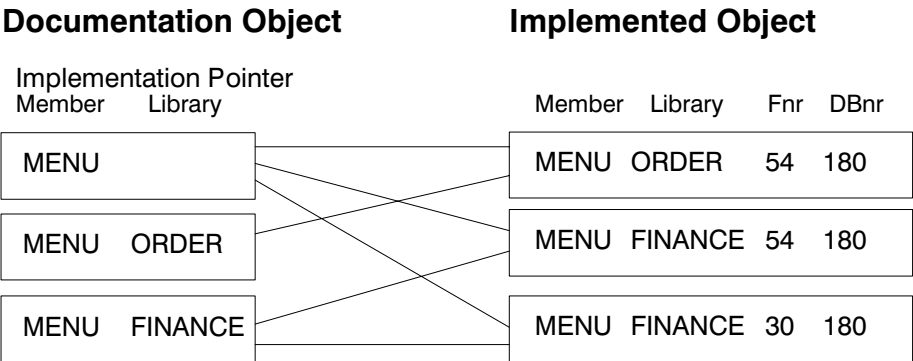
The table below lists the program types permitted for a program written in a particular language and indicates whether the program can have a list of entry points. In third generation languages, marked * in the table below, functions and subprograms can be documented as programs of type *F* and *N* respectively, but any active references for these programs will have type *P* (main program). The active references of these programs will be correctly connected in the active retrieval functions to programs of types *P*, *N* and *F*.

Language		Permitted Program Types	Entry Points allowed?
B	BAL (Assembler)*	C D F I N P	yes
C	COBOL*	C D F I N P	yes
E	Natural EL	D Y	no
F	FORTRAN*	C D F I N P	yes
G	ADA*	C D F N P	yes
H	C*	C D F I N P	yes
J	Job Control Language	D J	no
N	Natural	A C D G H I L M N O P S T 4 5	no
O	Other	C D F H M N P	yes
P	PL/I*	C D F I N P	yes
Q	Static SQL	D I P	yes
S	SQL procedure language	R	no
Z	System program	D E	yes
0 – 9	<i>user-defined</i>	C D F N P	no

New languages (code 0 – 9) are defined with the program U-PGMLAN. See Chapter **User Exits** in the *Predict Administration Manual*.

Combinations of Parameters for Natural Programs

If the same member is used in several libraries, multiple documentation of this member can be avoided by omitting parts of the implementation pointer. Predict then finds out for itself all the libraries in which this member exists.
 In the example below, the library name is omitted.



The valid combinations of implementation pointer parameters permitted for Natural programs are shown below.

Member	✓	✓	✓	✓
Library		✓	✓	✓
Fnr			✓	✓
DBnr				✓

Program List Specific Editor Commands

The following commands are available when editing one of the following lists:

- File list
- Entry point list
- Program list.

ACTIVE

Insert information from Xref data into the object list. Mark objects that are used with *<active*, and mark objects that are not used with *<unused*.

Xref data without a corresponding documentation object is marked **NOT DOCUMENTED**. An object ID can then be entered and the *.E* command can be used to create a Predict object corresponding to the Xref data. The implementation pointer for the new object is derived from Xref data and automatically inserted into the input fields of the *Add* menu.

UPDATE

Update active reference data in the object list.

Mark used objects with *<active* and delete unused objects from the list.

Comments on the ACTIVE command (above) also apply to this command.

RESET

Switches back to normal edit mode after ACTIVE or UPDATE have been issued. Information displayed in the right column is no longer derived from Xref data but is taken from the Predict objects. All lines marked **NOT DOCUMENTED** are removed from the list.

X and Y marks and scan values specified with the SCAN command are reset (as with the RESET command in any other list editor).

Generating Database Request Modules (DBRMs) from Program Objects of Language *Q* (Static SQL)

A Natural for DB2 database request module (DBRM) can be generated by the function *CREATE DBRM* of Natural DB2 from the list of entry points in a Predict object for a program of language *Q* (static SQL).
 See Chapter **DB2 and SQL/DS** in the Manual *Predict and Other Systems*.

Each entry point must be a Natural program that uses this DBRM. The Predict object should specify the member where the function *GENERATE DBRM* is to store the DBRM. The table below lists the columns of information that can be stored about entry points for a program of language *Q* only.

Column	Meaning
NAT-lib	The name of the library in which the Natural program is stored.
NAT-pgm	The name of the member in which the Natural program is stored.
Typ	The subtype of the Predict object for the Natural program.
Documentation	The ID of the Predict object for the Natural program.

The name of each entry point is concatenated. For detailed information on how the name is created see Chapter **Static SQL** in the Manual *Predict and Other Systems*. This name is used for the entry point when displaying the DBRM's Predict definition (retrieval function) or its active references (LIST XREF command).

For any type of program except *Q*, the names of the entry points are stored in a single column. The editor commands ACTIVE and UPDATE can be used to insert active reference data into an entry point list.

Purge Program (Code P)

The following rules apply:

- A program cannot be purged if it is linked to packagelist.
- If a program is implemented, a message tells you that Xref data will be deleted, too.

Redocument Program (Code X)

Creates Predict documentation objects for implemented programs (members). The function is used when redocumenting applications.

Predict retrieves the information needed to create the documentation object for a member either by scanning source code (only for Natural programs) or by evaluating Xref data.

Calling a Redocument Function

Online

Redocument functions are executed in two steps:

1. Select the programs to be processed using the parameters in the first *Redocument Program* screen as selection criteria. See page 358.
2. Determine the scope of the redocumentation using parameters in the second *Redocument Program* screen. See page 361.

Batch Mode

The function *Redocument program* is one of the few maintenance commands that can be entered in batch mode. The additional parameters that can be specified and a sample REDOCUMENT command are given in Chapter **Predict in Batch Mode** in the *Predict Administration Manual*.

Redocumenting Programs under Natural Security

Under Natural Security, some restrictions apply to this function to prevent unauthorized access to Natural sources. The same logic is used as in the SYSMAIN utility to check the user's access rights. The switch SYSMAIN from Natural Security is also interpreted in the *Redocument program* function.

See the *Natural Security Manual*, Chapter **SYSMAIN under Natural Security** for more information.

Selecting Programs to be Redocumented

Enter code *X* in the *Program Maintenance Menu* to display the *Redocument Program* screen:

12:25:12
Plan 0

***** P R E D I C T 4.1.1 *****
- Redocument program -

1999-03-03

Member
Library
Language* N
Source/Xref S (S/X)
Member types*

NATURAL
Source
All

File number 54
Database number ... 180
Password
Cipher

Parameters

Member name	Name of the member to be redocumented. Asterisk notation can be used.
Library	Depending on the parameter <i>Source/Xref</i> , either <ul style="list-style-type: none"> - library containing the members to be redocumented are stored, or - library of Xref data.
Language	Language of the program. Valid values: <ul style="list-style-type: none"> B BAL/Assembler C COBOL F FORTRAN G Language ADA H Language C N Natural P PL/I Q Static SQL X All but Natural <p>If option <i>X (All but Natural)</i> is entered, Predict redocuments all 3GL programs that meet the specified selection criteria.</p>

Parameters

Source/Xref	<p>S Source code is evaluated to create the Predict object (only for Natural programs).</p> <p>X Xref data is evaluated.</p>
Member type	<p>Additional selection criterion. Only member of the given types will be redocumented. For Natural programs, the following types can be specified:</p> <p>A Parameter Data Area</p> <p>C Copy Code</p> <p>G Global Data Area</p> <p>H Help routine</p> <p>L Local Data Area</p> <p>M Map / Help map</p> <p>N Subprogram</p> <p>O Natural command processor</p> <p>P Program</p> <p>S Subroutine</p> <p>Y Natural expert model</p> <p>4 Class</p> <p>5 Resource</p> <p><i>blank,*</i> All types</p> <p>A list of up to 9 member types can be specified. Member types can be specified without any delimiter (for example: <i>ACFH</i>)</p> <p>For third generation languages, only <i>P</i> (program) can be specified.</p>
File number, Database number	<p>Specify the FUSER file where the members to be processed are stored.</p> <p>Only applicable if <i>Source/Xref</i> is set to <i>S</i>.</p>
Password, Cipher	<p>Password and cipher code defined in Adabas can be specified if required.</p> <p>Only applicable if <i>Source/Xref</i> is set to <i>S</i>.</p>

Specifying the Redocument Parameters

The following screen appears if language type *N* is entered in the *Redocument Program Menu*.

```

14:17:28          ***** P R E D I C T  4.1.1  *****          1999-11-18
Plan    0          - Redocument program -

Processing options
Processing option .....* L List only
Link to system ..... GER-SY
Library structure .....*

Naming options (only applicable if 'Processing option' is 'Add')
Program ID prefix .....
Lib.name as sec.prefix ..... Y (Y,N)

Contents of documentation                                Implementation pointer
Abstract .....* S Statistics                            Library .. Y (Y/N)
Description .....* B Header comment                    Fnr ..... Y (Y/N)
Replace/append description .. R (R/A)                   DBnr ..... Y (Y/N)
Program list .....* U Update
File list .....* U Update
Default owner .....
First default keyword .....
Second default keyword .....
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Next Stop Last LnkEl Flip Print Impl AdmFi SelFi Prof Main

```

Using Default Values

All parameters of the *Redocument* function can be set to default values by the data dictionary administrator in the *Redocumentation Using Source Code* screen, which is called by code *R* in the *Modify General Defaults Menu*. Individual input fields can be protected. Protected default values cannot be overwritten. Protected fields are skipped when positioning the cursor with TAB.

Note:

Depending on the parameter *Source/Xref* (see page 359), some parameters may not be contained in the screen. This is stated in the description of parameters below.

Parameters

Processing options

Processing option

Determines the type of processing performed by the redocument function. Valid values:

- A *Add*: creates Predict objects for programs that are not already documented. Programs that meet the given selection criteria and are already documented will also be contained in the list and marked with the string *Rejected* in the column *Status*.
- R *Add and Replace*: creates documentation for all programs that meet the given selection criteria. Existing documentation objects will then be replaced.
If the program is a class, the objects of type interface, method and property are also documented.
- L *List*: lists programs that have not yet been documented in Predict. Programs that meet the given selection criteria and are already documented will also be contained in the list and will be marked with string *Rejected* in the column *Status*.

Link to system

If *Processing option* is set to *Add* or *Replace*, the program objects created by the function are automatically linked as children to the specified System.

If no system is specified, Predict looks for a documentation object of type *system* with the given *Library*, *DBnr* and *Fnr*. The system object that meets most of these criteria is inserted in this input field.

Structure

You can specify a library structure that is used to determine the used programs.

If no structure is specified, evaluation is performed without a library structure.

Naming options	Only applicable if <i>Processing option=Add</i> . The ID of the program object created by the redocument function contains up to three parts, separated by hyphens:
Program ID prefix	A prefix that can be specified with the parameter <i>Program ID prefix</i> .
Lib.name as sec.prefix	The library name of the member as secondary prefix, if the parameter <i>Lib.name as sec.prefix</i> is set to <i>Y</i> . The third part is the name of the member.
Contents of Documentation	These parameters determine the information to be documented.
Abstract	Determines the information to be contained in the abstract: S Statistical data (including the date and time when a member was cataloged). B Comment lines in the header. Only applicable if <i>Source/Xref</i> is set to <i>S</i> . A Comment lines in the header and statistical information. Only applicable if <i>Source/Xref</i> is set to <i>S</i> . N No abstract is created.
Description	Determines the information to be included in the extended description. Only applicable if <i>Source/Xref</i> is set to <i>S</i> . A Comment lines. Start with either * or /* in the first column followed by a series of characters other than * or <i>blank</i> . B All comment lines in the header of the member. R Comment lines and remarks. A remark starts with /* in any column and is followed by a series of any non-blank characters. S The whole source program. N No extended description is created.
Replace/append descr.	Determines handling of extended descriptions. Only applicable if <i>Source/Xref</i> is set to <i>S</i> . A The extended description of a Predict program object that is replaced (see <i>Processing option</i> above) is not overwritten. Instead, the new extended description is appended to the old extended description.

	<p>R The old extended description is overwritten when a Predict object is replaces. Default.</p>
Program list	<p>Programs that are called from within a program (for example via a CALL or FETCH statement) can be included in the program list of the object.</p> <p>The parameter <i>Program list</i> has the following options:</p> <p>U <i>Update</i>. The old contents of the program list are completely replaced by the information extracted from the Xref data.</p> <p>A <i>Add active links</i>. Additional entries in the program list are created, documenting the use of programs not already documented. All other entries in the list will be kept. This option only makes sense if an existing documentation object is replaced.</p> <p>N No entries in the program list are created.</p>
File list	<p>Files that are used by a program can be included in the file list of the program.</p> <p>See <i>Program list</i> above for description of the options.</p>
Default owner	<p>The default Owner specified is included in the owner list of the object. The Owner must be defined in at least one object of type <i>user</i>.</p> <p>Only applicable if <i>Processing option</i> is set to A.</p>
First default keyword	<p>Only applicable if <i>Processing option</i> is set to A.</p>
Second default keyword	<p>Two keywords can be specified that are linked to the objects created. The keywords must be defined in Predict.</p>

Implementation pointer
Library, DBnr, Fnr

These parameters determine two things:

- The amount of information to be stored in the implementation pointer of the Predict program object to be created by the *Redocument* function. If the library, DBnr or Fnr is to be added the implementation pointer by the redocument function, the respective parameter must be set to *Y*.
- Which information of existing Predict program objects is evaluated to determine whether an implemented program is already documented.

For example: if *Library*, *DBnr* and *Fnr* are set to *Y*, a Predict object is only regarded as the documentation of an implemented program if its implementation pointer contains correct values for the following:

- member name
- library
- DBnr
- Fnr.

Note: If the parameters *Implement. Library* and *Implement. DBNR/FNR* in the Predict *Defaults* have been set to either *Disallow (D)* or *Force (F)*, the parameters above cannot be set to *Y* or *N* respectively.

Handle */** in columns 1+2 as comment or as remark

- C A line with */** in the first two columns is interpreted as a comment line.
- R A line with */** in the first two columns is interpreted as a remark.

Edit procedure code of a program (Code Y)

This function can only be executed for programs of type *SQL procedure* with language *SQL procedure*.

The Predict Description Editor is called. Additional checks are performed when the procedure code is cataloged. See Chapter **Editors in Predict** in the *Predict Reference Manual* for more information.

Program Retrieval

Program-Specific Retrieval Parameters

The following program-specific parameters determine the scope of reports.

Program of type	Restrict report to programs of the given type. See page 343 for a list of valid types.
Language	Restrict report to programs of the given language. See page 343 for a list of valid languages.
Member	Restrict report to programs documenting the given member.
Library	Restrict report to programs documenting a member in the given library. See page 344 for a list of standard libraries.
User system Fnr	Restrict report to program objects documenting implemented programs in this user system file.
User system DBnr	Restrict report to program objects documenting implemented programs in this database.
in system	Restrict report to programs linked to this system object.

Function *Programs with Children* with Child Type *Program*

As of this version of Predict, the Retrieval function *Program with Children* with child type *Program* evaluates only documentation data. If you need information on the implementation of a program, use the new Active Retrieval function *Programs using programs*.

Layout of Program Lists

14:54:04		***** P R E D I C T 4.1.1 *****				1999-04-24	
		- List Program -					
Cnt	Program ID	Type	Lang	Member	Library	Fnr	DBnr
14	STK-PR-O	P	O	CP1E	ST-PRDE		
15	STK-PR-STATIC-SQL	P	Q	HUXEL	HUXEL	255	255
16	STK-PR-1	O					
17	* STK-PR-2	N	N	N-SECCHC			
		Implementation: N-SECCHC GMA				54	180
		N-SECCHC NEWDICCO				54	180
18	STK-PROC	R	S	KSTK	KKKKK		
19	* STK-REDOC	P	N	Z-HI1	STK		
		Implementation: Z-HI1				54	180

Meaning of Columns

Program ID	The ID of the program object.
<i>Note:</i>	An asterisk in the first column indicates that the program is implemented. <i>Implemented</i> in this sense means that Xref data exists for the documentation object.
Type	Program type. See page 343 for a list of valid types and codes.
Lang.	The language in which the program is written. See page 343 for list of valid languages and codes.
Member, Library, Fnr, DBnr	Implementation pointer of the program object, or – if the object is implemented – the physical implementation of the member(s) documented by the program. In the sample screen above, program STK-PR-2 has implementation pointer N-SECCHK (for member) and documents member N-SECCHK in libraries GMA and NEWDICCO.

Output Options for Program Retrieval

Retrieval Type	D				B				O				T							
													dummies=Y N				dummies=D P			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																				
Adabas sizes																				
Association attributes					✓	✓	✓	✓					✓	✓	✓	✓				
Attributes	✓				✓				✓				✓				✓			
Check expression																				
Composed fields																				
Connecting character						✓								✓						
Cover page	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Description	✓				✓	✓			✓				✓	✓			✓			
Display length																				
Display modifier	✓				✓				✓				✓				✓			
Dummy/Placeholder														✓		✓		✓		✓
DV-field expression																				
Entry points	✓				✓				✓				✓				✓			
Extract	✓				✓	✓			✓				✓	✓			✓	✓		
Generation layout																				
Adabas version																				
Language																				
Alignment/sync.																				
Position/Offset																				
Counter length																				
Compiler																				
Replace with syn.																				

Retrieval Type	D				B				O				T							
													dummies=Y N				dummies=D P			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓				✓	✓			✓				✓	✓			✓			
Linked verification																				
Mark implementation	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
No. abstract lines	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
Natural options																				
Owner	✓				✓	✓			✓				✓	✓			✓			
With users	✓				✓	✓			✓				✓	✓			✓			
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Procedure code	✓				✓				✓				✓				✓			
Rules																				
Show implementation	✓		✓		✓				✓				✓				✓			
Sorted by field																				
Subquery																				
Synonyms																				
STARTAB elements																				
Trigger																				
Use Con-form	✓				✓	✓			✓				✓	✓			✓			
User exit	✓				✓				✓				✓				✓			
3GL specification																				

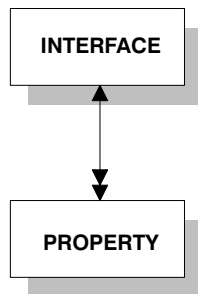
Output Options for Program Retrieval (Continued)

Retrieval Type	U				E				C			
Output Mode	D		L		T		X		L		D	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes												
Adabas sizes												
Association attributes					✓	✓						
Attributes	✓					✓		✓				
Check expression												
Composed fields												
Connecting character						✓		✓				
Cover page	✓		✓		✓	✓	✓	✓	✓	✓		
Description	✓							✓				✓
Display length												
Display modifier	✓											
Dummy/Placeholder						✓		✓	✓		✓	
DV-field expression												
Entry points	✓											
Extract	✓					✓		✓			✓	✓
Generation layout												
Adabas version												
Language												
Alignment/sync.												
Position/Offset												
Counter length												
Compiler												
Replace with syn.												

Retrieval Type	U				E				C			
Output Mode	D		L		T		X		L		D	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓					✓		✓				✓
Linked verification												
Mark implementation	✓		✓		✓	✓	✓	✓		✓		✓
No. abstract lines	✓		✓			✓		✓		✓		✓
Natural options												
Owner	✓					✓		✓				✓
With users	✓											✓
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Procedure code	✓											
Rules												
Show implementation	✓											
Sorted by field												
Subquery												
Synonyms												
STARTAB elements												
Trigger												
Use Con-form	✓							✓				✓
User exit	✓											
3GL specification												

PROPERTY

This object type is used to document the *properties* of an *interface*.



In the Predict meta structure, a *property* can have a parent of type *interface*.

How this Chapter is Organized

- **The Property Maintenance Menu**, page 373
 - The *Add a Property* screen, page 374
- **Property Retrieval**, page 375

The Property Maintenance Menu

This menu is called with function code *M* and object code *PY* in a Predict main menu, or with the command MAINTAIN PROPERTY.

11:23:08
Plan 0

***** P R E D I C T 4.1.1 *****
- (PY) Property Maintenance -

1999-09-30
Profile SYSTEM

Function	Function
A Add a Property	D Display Property
C Copy Property	L Link children
M Modify Property	O Edit owners of a Property
N Rename Property	S Select Property from list
P Purge Property	W Edit description

Function a

Property ID example

Copy ID

In Interface

Restrictions*

Profile Default,empty

Child type*

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---

- Next Stop Last LnkEl Flip Print Impl AdmFi Selfi Prof Main

Parameters

The *Property Maintenance Menu* contains only global attributes. See page 6.

The functions are described in Chapter **Maintenance** in the *Predict Reference Manual*.

The *Add a Property* Screen

The following screen appears for the function *Add a Property*. The screens for functions *Copy* and *Modify* are similar.

11:25:10

***** P R E D I C T 4.1.1 *****

1999-09-30

- Add a Property -

Property EXAMPLE

in Interface ...*

Keys ..

Zoom: N

Attributes

Property name

Readonly (Y/N)

Abstract Zoom: N

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Parameters

Property	ID of the property.
Property name	Name of the property.
Readonly	Y Variables cannot be modified.

Property Retrieval

Information on property objects is gathered using standard retrieval functions. See Chapter **Retrieval** in the *Predict Reference Manual*.

Output Options for Property Retrieval

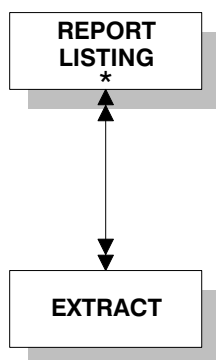
The output options valid for this object type are identical to those for object type *dataspace*. See page 58.

REPORT LISTING

Objects of type *report listing* log

- a transfer operation of the Predict Coordinator, or
- a conversion operation.

Report listings are added automatically with an ID assigned by the system. For this reason, the functions *Add*, *Copy* and *Rename* are not available for this object type.



In the Predict metastructure, a *report listing* can have parents and children of the following types:

Valid parents: *User-defined*

Valid children: Extract (default child)

When transferring data with the Predict coordinator, the extract containing the objects to be transferred is automatically linked as a child to the report listing.

See the *Predict Coordinator Manual* for more information.

How this Chapter is Organized

- **Report Listing Maintenance**
 - *Report Listing Maintenance Menu*, page 377
 - The *Modify Report Listing* screen
- **Report Listing Retrieval**
 - Layout of Report Listing Lists, page 380
 - Output Options, page 380

The Report Listing Maintenance Menu

This menu is called with function code *M* and object code *RT* in a Predict main menu or with the command MAINTAIN REPORTLISTING.

11:24:44
Plan 0

***** P R E D I C T 4.1.1 *****
- (RT) Report listing Maintenance -

1999-08-04
Profile JCA

Function

Function

M Modify Report listing
P Purge Report listing

D Display Report listing
L Link children
O Edit owners of a Report listing
S Select Report listing from list
W Edit description

Function

Report listing ID

Copy ID

Restrictions*

Profile JCA ,used

Child type*

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---

Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next

Parameter

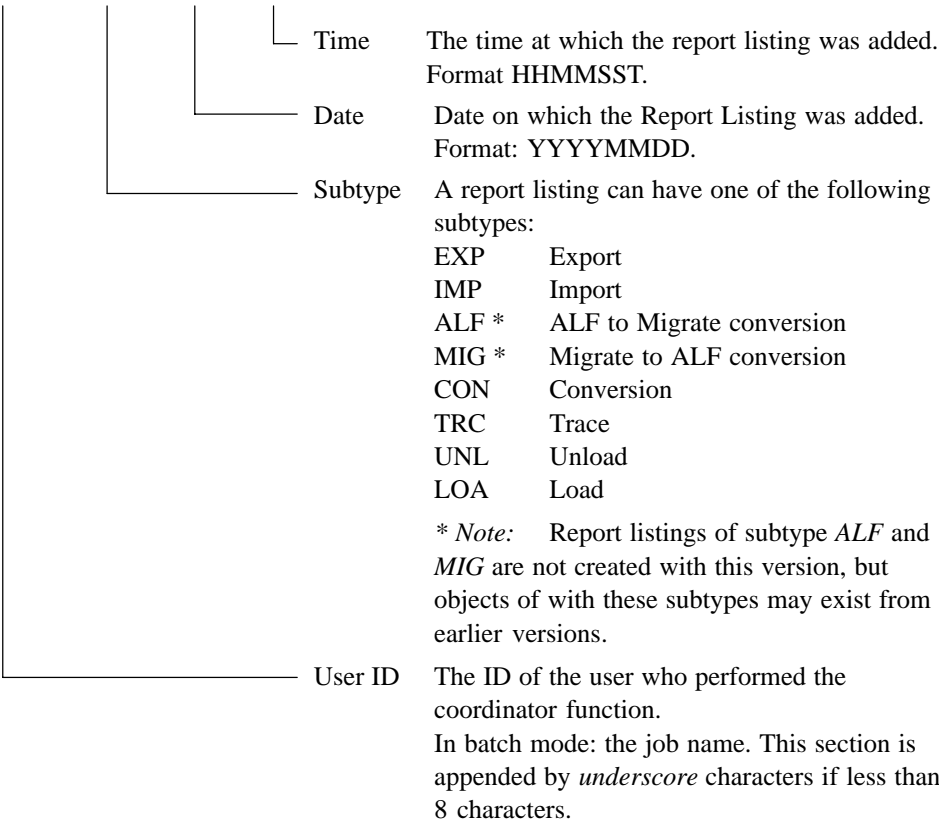
Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Function Enter one of the codes in the menu to execute one of the functions. These functions are described in Chapter **Maintenance** in the *Predict Reference Manual*.

Note: As report listings are added automatically with an ID which is assigned by the system, the functions *Add*, *Copy* and *Rename* are not available for this object type.

Report Listing ID ID of the report listing object. This ID is assigned automatically when the object is added and is composed as follows:

USR____-TYP-19940803-1522453



Copy ID Not applicable to this object type. See note on page 377.

Modify Report Listing Screen

11:35:13

***** P R E D I C T 4.1.1 *****

1999-02-25

- Modify Report listing -

Report listing .. BOE_____EXP-19950214-1555101

Added 1998-02-14 at 15:55

by EXPORT

Zoom: N

Keys ..

Attributes

Subtype Export

Processing

Exported

Not Exported

3

0

Abstract

Zoom: N

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Explanation

Added by	The function with which the report listing was added. The user who created the report listing can be seen in the first eight characters of the report listing ID.
Subtype	Subtype of report listing. Conversion (ALF to Migrate or Migrate to ALF conversion) Export Import Trace Load Unload
Processing	
Exported / Not Exported	For the function <i>Export</i> : The number of objects successfully exported / objects not exported due to errors.
Note:	See the extended description of the report listing for a complete list of these objects.

Explanation

Loaded / Replaced / Not Loaded

For the function *Import*: The number of new objects successfully loaded / existing objects overwritten / objects not loaded due to errors.

Note: See the extended description of the report listing for a complete list of these objects.

Report Listing Retrieval

Information on report listings is retrieved using standard retrieval functions. These functions are described in Chapter **Retrieval** in the *Predict Reference Manual*.

Layout of Report Listing Lists

11:01:24

***** P R E D I C T 4.1.1 *****

1999-02-25

- List Report listing -

Cnt	Report listing ID	Subtype
1235	FH_____ -EXP-19950213-1133434	Export
1236	FH_____ -EXP-19950213-1134044	Export
1237	FH_____ -IMP-19950213-1135086	Import
1238	FH_____ -IMP-19950213-1750037	Import
1239	FH_____ -IMP-19950213-1758171	Import
1240	FH_____ -MIG-19950209-1531474	Convert
1241	GER-RT	Import
1242	GER_____ -ALF-19950206-1017009	Convert

Output Options for Report Listing Retrieval

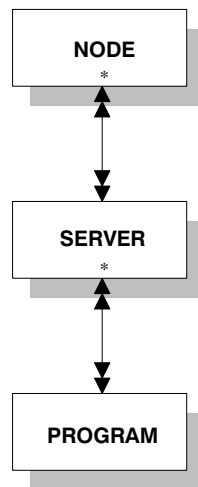
The output options valid for this object type are identical to those for object type *dataspace*. See page 58.

SERVER

This object type, together with object type *node*, is used to document remote procedure calls.

An object of type *server* documents all programs available on a logical server.

In the Predict metastructure, a *server* can have parents of type *node* and children of type *program*.



How this Chapter is Organized

- **Server Maintenance**
 - The Server Maintenance Menu, page 383
 - The *Add a Server* Screen, page 384
- **Server Retrieval**
 - Layout of Server Lists, page 385

The Server Maintenance Menu

This menu is called with function code *M* and object code *SV* in a Predict main menu or with the command MAINTAIN SERVER.

```
12:54:29          ***** P R E D I C T  4.1.1  *****          1999-10-04
Plan  10          - (SV) Server Maintenance -                      Profile JCA

Function                                Function

A  Add a Server                        D  Display Server
C  Copy Server                        L  Link children
M  Modify Server                      O  Edit owners of a Server
N  Rename Server                     S  Select Server from list
P  Purge Server                      W  Edit description

Function .....
Server ID .....
Copy ID .....
In Node ..... JCA-N01

Restrictions .....*   Profile JCA ,used           Child type ....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Menu  Canc  S-fi  E-el  M-pr  Print Impl  Last  FLIP  PROF  Next
```

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Parameters

Function Enter one of the codes from the menu to execute one of the maintenance functions. These functions are described in Chapter **Maintenance** in the *Predict Reference Manual*.

The *Add a Server* Screen

The following screen is called for function *Add a Server*. The screens for functions *Copy* and *Modify* are similar.

13:29:37

***** P R E D I C T 4.1.1 *****

1999-10-04

- Add a Server -

Server JCA-SV1

in Node* JCA-NO1

Keys ..

Zoom: N

Server name

Abstract Zoom: N

.

.

.

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Parameters

Server ID	ID of the server object.
in Node	ID of the parent node.
Server name	Name of the server must be specified. Up to 8 characters.

Server Retrieval

Only standard retrieval functions are used. See Chapter **Retrieval** in the *Predict Reference Manual*.

Layout of Server Lists

Server lists contain the server IDs and the server names.

11:19:16	***** P R E D I C T 4.1.1 *****	1999-02-28
	- List Server -	

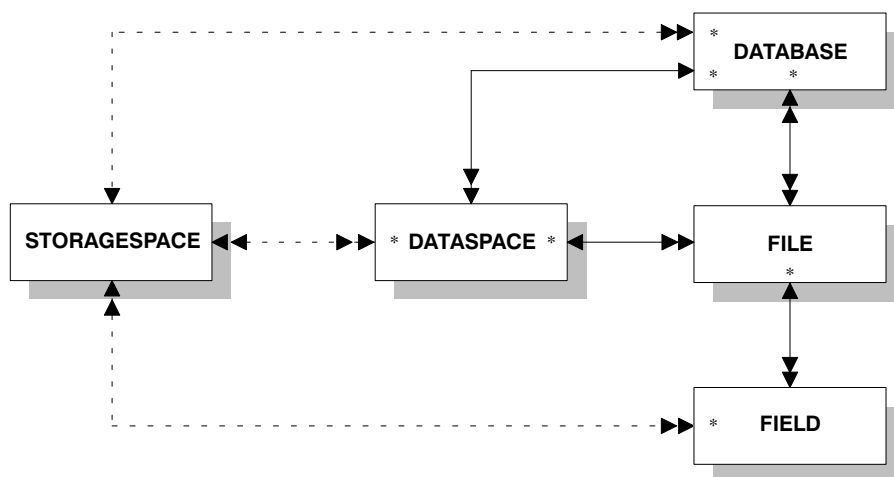
Cnt	Server ID	Server name
1	FST-SERVER	FST-TEST
2	HEB-SV	SERVSERV
3	JCA-SV1	server1

Output Options for Server Retrieval

The output options valid for this object type are identical to those for object type *Dataspace*. See page 58.

STORAGESPACE

DB2 storagegroups are documented in Predict with the object type *Storagepace*. See Chapter **DB2 and SQL/DS** in the Manual *Predict and Other Systems*.



In the predefined Predict metastructure, a *storagepace* has no standard link. References to storagespaces are realized with the attribute (*Default*) *Storagepace* of objects of type *database*, *dataspace* and *field*.

How this Chapter is Organized

- **Storagepace Maintenance**
 - The *Storagepace Maintenance Menu*, page 387
 - The *Add/Copy/Modify Storagepace* Screen, page 388
 - Function *Purge Storagepace*, page 389
- **Storagepace Specific Retrieval**
 - Function *Unused Storagepaces*, page 390
 - Layout of Storagepace Lists, page 390
 - Valid Output Options, page 391

The StorageSpace Maintenance Menu

The *StorageSpace Maintenance* menu is called with function code *M* and object code *SC* in a Predict main menu or the command MAINTAIN STORAGESPACE.

10:22:43
Plan 0

***** P R E D I C T 4.1.1 *****
- (SC) StorageSpace Maintenance -

1999-03-06
Profile JCA

Function

Function

A Add a storagespace
C Copy storagespace
M Modify storagespace
N Rename storagespace
P Purge storagespace

D Display storagespace
L Link children
O Edit owners of a storagespace
S Select storagespace from a list
W Edit description of a storagespace

Function

StorageSpace ID ..
Copy ID

Restrictions* Profile JCA ,used Child type*

Command ==>

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next

Parameters

Note:

Parameters not listed here are described under **Global Attributes**, page 6.

Function Standard maintenance functions are described in Chapter **Maintenance** in the *Predict Reference Manual*. Function *Purge* is described on page 389

The Add a Storagepace Screen

The screen is displayed for the *Add a Storagepace* function. The *Copy* and *Modify* screens are similar.

10:26:41

***** P R E D I C T 4.1.1 *****

1999-03-06

- Add a Storagepace -

Storagepace JCA-SC1

Keys ..

Zoom: N

Storagepace attributes

Storagegroup name

VSAM catalog name

Password required N (Y,N)

Device type

Abstract

Zoom: N

Volumes

1

7

13

19

25

31

37

43

49

EDIT: Owner: N Desc: N

MORE Volumes: N

Parameters

Note:

Parameters not listed here are described under **Global Attributes**, page 6.

Storagepace	ID of the Predict storagepace object.
Storagepace attributes	
Storagegroup name	Name of the storagegroup in DB2.
VSAM catalog name	Name or alias of an ICF catalog. Aliases are used for names of ICF catalogs that are longer then eight characters.
Password required	Y Access to the specified ICF catalog is protected with a password.
Device type	For documentation purposes.

Volumes

Physical volume(s) where the storagespace resides.

Up to 55 volumes can be entered here. Specify *MORE volumes=Y* to specify up to 140 volumes.

StorageSpace-Specific Maintenance

Purge StorageSpace (Code *P*)

The following restriction applies to this function:

- A storagespace cannot be deleted if it is still referenced by a database, a dataspace or a file.

Otherwise this function behaves as described in Chapter **Maintenance** in the *Predict Reference Manual*.

Storagepace Retrieval

Unused Storagepaces (Code N)

Lists unused Storagepaces. A storagepace is regarded to be unused if it is not referenced in a dataspace or field object.

Layout of Storagepace Lists

10:56:33

***** P R E D I C T 4.1.1 *****

1999-03-06

- List Storagepace -

Cnt	Storagepace ID	Stgr name	VCAT name	DB2	Pw.req
1	ARH-SC	ARH_SC			
2	ARH-SC-2	STOGR2			
3	BOE-ST01	FRITZ			
4	CHD-SC	YYYY			
5	* CHD-STORAGESPACE	CHDSPC	PB4		
6	CHD-STOSPACE	SPATZ			
7	* CHD-STOSPATZ	CHDSPTZ	PB4		
8	* DEVELOP	DEVELOP	DB2V23		

Meaning of Columns

Storagepace ID	ID of the storagepace. If the output option <i>Mark implementation</i> is set to <i>Y</i> , implemented objects are marked with an asterisk. ‘ <i>Implemented</i> ’ means here that a DB2 storagegroup has been generated from the storagepace.
Stgr name	Name of the DB2 storagegroup.
VCAT name	Name or alias of an ICF catalog.
DB2 Pw.req	Y Access to the specified ICF catalog is protected with a password.

Output Options for StorageSpace Retrieval

Retrieval Type	D				B				O				T							
													dummies=Y N				dummies=D P			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																				
Adabas sizes																				
Association attributes					✓	✓	✓	✓					✓	✓	✓	✓				
Attributes	✓				✓				✓				✓				✓			
Check expression																				
Composed fields																				
Connecting character						✓								✓						
Cover page	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Description	✓				✓	✓			✓				✓	✓			✓			
Display length																				
Display modifier	✓				✓				✓				✓				✓			
Dummy/Placeholder														✓		✓		✓		✓
DV-field expression																				
Entry points																				
Extract	✓				✓	✓			✓				✓	✓			✓	✓		
Generation layout																				
Adabas version																				
Language																				
Alignment/sync.																				
Position/Offset																				
Counter length																				
Compiler																				
Replace with syn.																				

Retrieval Type	D				B				O				T							
													dummies=Y N				dummies=D P			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓				✓	✓			✓				✓	✓			✓			
Linked verification																				
Mark implementation	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
No. abstract lines	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
Natural options																				
Owner	✓				✓	✓			✓				✓	✓			✓			
With users	✓				✓	✓			✓				✓	✓			✓			
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Procedure code																				
Rules																				
Show implementation	✓				✓				✓				✓				✓			
Sorted by field																				
Subquery																				
Synonyms																				
STARTAB elements																				
Trigger																				
Use Con-form	✓				✓	✓			✓				✓	✓			✓			
User exit	✓				✓				✓				✓				✓			
3GL specification																				

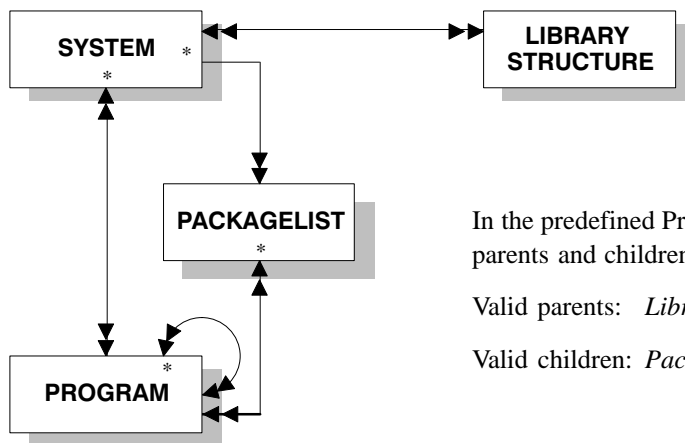
Output Options for Storagespace Retrieval (Continued)

Retrieval Type	U				E				C				N			
Output Mode	D		L		T		X		L		D		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																
Adabas sizes																
Association attributes					✓	✓										
Attributes	✓					✓		✓					✓			
Check expression																
Composed fields																
Connecting character						✓		✓								
Cover page	✓		✓		✓	✓	✓	✓	✓	✓			✓		✓	
Description	✓							✓				✓	✓			
Display length																
Display modifier	✓												✓			
Dummy/Placeholder						✓		✓								
DV-field expression																
Entry points																
Extract																
Generation layout																
Adabas version																
Language																
Alignment/sync.																
Position/Offset																
Counter length																
Compiler																
Replace with syn.																

Retrieval Type	U				E				C				N			
Output Mode	D		L		T		X		L		D		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓					✓		✓					✓			
Linked verification																
Mark implementation	✓		✓		✓	✓	✓	✓		✓			✓		✓	
No. abstract lines	✓		✓			✓		✓		✓		✓	✓		✓	
Natural options																
Owner	✓					✓		✓					✓			
With users	✓												✓			
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓				✓		✓	
Procedure code																
Rules																
Show implementation	✓												✓			
Sorted by field																
Subquery																
Synonyms																
STARTAB elements																
Trigger																
Use Con-form	✓							✓				✓	✓			
User exit	✓												✓			
3GL specification																

SYSTEM

An application can be documented with a Predict object of type *System*. See page 398 for a list of possible system types.



In the predefined Predict metastructure, a *System* can have parents and children of the following types:

Valid parents: *Library Structure*

Valid children: *Packagelist*, *Program* (default child)

How this Chapter is Organized

- **System Maintenance**
 - The *System Maintenance* Menu, page 397
 - The *Add/Copy/Modify System* Screen, page 399
 - Identifying Systems, page 400
 - Function *Purge System*, page 400
 - Function *Rename System*, page 400
- **System Retrieval**
 - System-Specific Retrieval Parameters, page 401
 - Function *Systems with Children* with Child Type *Program*, page 401
 - Layout of System Lists, page 401

The System Maintenance Menu

The *System Maintenance* menu is called with function code *M* and object code *SY* in a Predict main menu or the command MAINTAIN SYSTEM.

```
14:51:33          ***** P R E D I C T  4.1.1  *****          1999-03-03
Plan    0          - (SY) System Maintenance -          Profile JCA

Function                                Function

A  Add a system                        D  Display system
C  Copy system                        L  Link children
M  Modify system                      O  Edit owners of a system
N  Rename system                     S  Select system from a list
P  Purge system                      W  Edit description of a system

Function .....

System ID ..... CHD-3GL                System of type ....*
Copy ID .....
Library .....
Restrictions ....*   Profile JCA ,used   User system Fnr ....
                                           User system DBnr ...
                                           Child type .....*

Command ===>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next
```

Parameters

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Function	Standard functions are described in Chapter Maintenance in the <i>Predict Reference Manual</i> . Function <i>Purge system</i> is described on page 400, function <i>Rename system</i> on page 400.
----------	--

Parameters

System of type	<p>For the <i>Select</i> function: a system type can be specified as a selection criterion.</p> <p>For the <i>Add</i> and <i>Copy</i> functions: the system type can be specified here. This type will be passed to the <i>add System or copy System</i> screen.</p> <p>Valid values:</p> <p>A Application</p> <p>C Conceptual. Used to outline the preliminary description of an application in the design phase.</p> <p>G 3GL Application</p> <p>P DB2 plan. Used to document a DB2 application.</p> <p><i>blank</i> all</p>
Child type	<p>For function <i>Link children</i>: Specify object type of children to be linked. Valid values are <i>Packagelist</i> and <i>user-defined</i>.</p>

The Add/Copy/Modify System Screen

The screen is displayed for the *Add a System* function. The *Copy* and *Modify* screens are similar.

14:54:46

***** P R E D I C T 4.1.1 *****

1999-03-03

- Add a System -

System ID JCA-SY3

Type *

Keys ..

Zoom: N

Implementation pointer

Library

User system Fnr ...

User system DBnr ...

DB2 Plan name

Abstract

Zoom: N

EDIT: Owner: N

Desc: N

Prog.list: N

Parameters

Note:

Parameters not listed here are described under **Global Attributes**, page 6.

System ID	The ID of the Predict system object. A read-only field.
Type	System type. Enter asterisk to display valid values or see list on page 398 above.
Implementation pointer	
Library	The name of the library. For type G: The library can not be changed if Xref data exists (the library is used by a 3GL program).
User system Fnr	The file number of the user system file (FUSER).
User system DBnr	The database number of the user system file.
DB2 plan name	Unique DB2 plan name. Only applicable to DB2 plans (systems of type P).

System-Specific Maintenance

Identifying Systems

Systems documented with Predict objects of type *System* can be identified with three parameters: library, file number and database number. The three possible combinations of these parameters are shown below.

Library	✓	✓	✓
File number		✓	✓
Database number			✓

Purge System (Code *P*)

The following rules apply to this function:

- A system of type *G* (3GL application) cannot be deleted if Xref data exist
- If you confirm the function with **DELETE**, the following objects are deleted:
 - the system object
 - all links to child objects
 - all links from parent objects
- If you confirm with **SCRATCH**, the following objects are deleted additionally:
 - Programs linked to the system
(programs that are linked to packagelists are not deleted)
 - all links to/from objects that are deleted together with the system
 - Xref data for the system (including DBRMs and system programs)
 - Xref data for scratched programs (parameter *Language* = *Ada*, *BAL*, *COBOL*, *FORTRAN*, *PL/I*, *Static SQL*).

Rename System (Code *N*)

Use this function to change the ID and/or type of a system object. The following restriction applies:

- You cannot change the type of a system of type *3GL application* for which Xref data exists.

System Retrieval

System-Specific Retrieval Parameter

System of type

Applicable to system retrieval. Limits the scope of the function to systems of the type specified. Enter asterisk for possible values or see list on page 398.

Systems with Children (Code *T*), with Child Type *Program*

With this version of Predict, the retrieval function *Systems with Children* (with child type *Program*) evaluates only documentation data. If you require information on an implemented system, use the new active retrieval function *System containing programs*.

Layout of System Lists

15:40:59	*****	P R E D I C T	4.1.1	*****	1999-07-22
		- List System -			Page: 1
Cnt	System ID	Type	Library	Fnr	DBnr
1	ADABAS	C			
2	ARH-LO	C			
3	* ARH-SYS	A	ARH		
4	ARH-SYS-P	P			

Meaning of Columns

System ID

ID of the system object.
If the output option *Mark implementation* is set to *Y* implemented objects are marked with an asterisk. 'Implemented' in this case means that Xref data exists for at least one program contained in a library documented by the system object.

Meaning of Columns

Type	The type of system. See list of valid types and codes on page 398.
Library, Fnr, DBnr	Information on where a system is implemented: Library, file number and database number of the user system file.

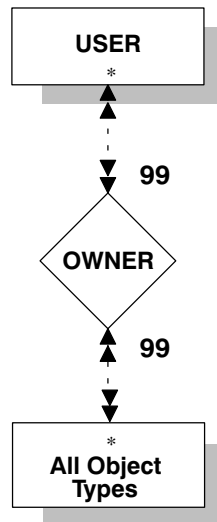
Output Options for System Retrieval

The output options valid for this object type are identical to those for object type *Dataspace*. See page 58.

USER/OWNER

The object type *user* contains information on users and organizational units, such as name, ID or position within the company.

One attribute of this object type is *owner*. Groups of users reflecting organizational units, such as project teams, can be formed by assigning individual users to an owner. Each user can belong to several owners. Owners can be associated to other types of Predict objects. See also **User and Keyword** in Chapter **Overview of Predict** in the Manual *Introduction to Predict*.



How this Chapter is Organized

- **User/Owner Maintenance**
 - The *User/Owner Maintenance Menu*, page 406
 - The *Add a User* Screen, page 408
- **User Specific Maintenance**, page 410
 - Purge User
- **User Retrieval**
 - User-Specific Retrieval Parameter, page 411
 - User-Specific Retrieval Functions
 - Users Related to Objects, page 411
 - Users Related to no Object, page 411
 - Layout of User Lists, page 411
 - Output Options for User Retrieval, page 412
- **Owner Maintenance**
 - Linking objects logically using Owners, page 416
 - Owner-specific Maintenance Functions, page 417
 - Rename/Merge Owner, page 417
 - Purge Owner, page 418
- **Owner Retrieval**
 - Owner-specific retrieval functions, page 419
 - Owners with no User, page 419
 - Objects with no Owners, page 420
 - Cross Reference Owners, page 420
 - Layout of Owner Lists, page 421
 - Output Options for Owner Retrieval, page 423

The User/Owner Maintenance Menu

The *User Maintenance* menu is called with function code *M* and object code *US* in a Predict main menu or the command MAINTAIN USER.

```
17:25:38          ***** P R E D I C T  4.1.1  *****          1999-02-24
Plan    4          - (US) User Maintenance -          Profile JCA

Function                                Function
A  Add a user                          L  Link children
C  Copy user                          O  Edit owners of an user
M  Modify user                        S  Select user from a list
N  Rename user                       W  Edit description of a user
P  Purge user                        R  Rename/Merge owner
D  Display user                      E  Purge owner

Function .....
User ID .....
Copy ID .....
User name .....
Owner .....
Restrictions ....*   Profile JCA ,used           Child type .....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next
```

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Parameters

Function	Executes one of the maintenance functions. Standard maintenance functions are described in Chapter Maintenance in the <i>Predict Reference Manual</i> . The function <i>Purge user</i> is described on page 410, <i>Rename/Merge</i> and <i>Purge owner</i> are described on page 416.
User ID	Identifier of the Predict user object. Must start with a letter, and can be up to 8 characters long. See also section Naming Conventions on page 6.

Parameters

User name	Name of the user. If the user name is unique, it can be specified instead of the user ID.
Owner	Owner ID. If the owner ID is unique, it can be specified instead of the user ID.
Child type	For function <i>Link children</i> : objects of this type are to be linked to the user. Valid values: <i>user-defined</i> .

The *Add a User* Screen

The screen is displayed for the *Add a user* function. The *Copy* and *Modify* screens are similar.

17:26:31
***** P R E D I C T 4.1.1 *****
1999-02-24

- Add a User -

User ID USR-123
Name
Keys ..
First Owner ID ..

Zoom: N

Business information
Function ..
Title
Organiz ...
Usage (ACC,UPD)

Phone
Extension ..
Mail code ..

User address
Street
Zip Code ..
State
Phone

No
City ..

Abstract

Zoom: N

EDIT: Owner: N Desc: N

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Parameters

User ID	ID of the user object.
Name	The name of the user
First owner ID	<p>The first owner to which the user belongs can be specified. More owners can be added to the owner list</p> <ul style="list-style-type: none"> - by entering <i>Y</i> in the <i>EDIT Owner</i> field. - with the <i>Edit owners of a user</i> function in the <i>User Maintenance</i> menu. <p>For a complete description see page 132.</p>

Parameters

User attributes	Various attributes describing the user’s position within the organization, telephone number and access privileges (parameter <i>Usage</i> with values <i>ACCEss</i> or <i>UPDate</i>). The attributes are used for documentation purposes only.
User address	Various address data for the user.

User Maintenance

Purge User

The following rules apply:

- If you confirm this function with **DELETE**, the following objects are deleted:
 - the user
 - all links to child objects
 - all links from parent objects
 - all sets created by this user
 - the workplan of the user
 - the Predict and LIST XREF profiles of the user
- A user will not be deleted with the **DELETE** option if
 - he is the only user in the user list of an owner and
 - this owner is assigned to an object where the option OWNER=FORCE has been defined in the metadata administration for this object type.
- If you confirm this function with **SCRATCH**, the following objects are deleted additionally:
 - All Owners assigned to the user are removed from the linked objects.
- A user will not be deleted with the **SCRATCH** option if
 - this would lead to all owners of an object being deleted and
 - the option OWNER=FORCE has been defined for this object type in the metadata administration.

User Retrieval

User-Specific Retrieval Parameter

User name

Limits the scope of the function to to users with the name specified.

User-Specific Retrieval Functions

Users Related to Objects (Code X)

Lists users and objects which are related to these users via an owner or a keyword.

Command: USED USER

Users Related to no Object (Code Y)

Lists users which are not related to any other objects in the data dictionary. The association between a user and a data dictionary object of any other type (except keyword) is always established indirectly through an owner, by associating the same owner with the user and with the other object.

Command: UNUSED USER

Layout of User Lists

```
17:30:17          ***** P R E D I C T  4.1.1  *****          1999-02-24
                        - List User -

-----
Cnt  User ID  User name
-----
  1  AAA      User1
  2  BBB
  3  CCC      User123
  4  DDD      DDD-TEST
```

Meaning of Columns

User ID ID of the user object.

User Name The name of the user.

Output Options for User Retrieval

Retrieval Type	D				B				O				T							
													dummies=Y N				dummies=D P			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																				
Adabas sizes																				
Association attributes					✓	✓	✓	✓					✓	✓	✓	✓				
Attributes	✓				✓				✓				✓				✓			
Check expression																				
Composed fields																				
Connecting character						✓								✓						
Cover page	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Description	✓				✓	✓			✓				✓	✓			✓			
Display length																				
Display modifier	✓				✓				✓				✓				✓			
Dummy/Placeholder														✓		✓		✓		✓
DV-field expression																				
Entry points																				
Extract	✓				✓	✓			✓				✓	✓			✓	✓		
Generation layout																				
Adabas version																				
Language																				
Alignment/sync.																				
Position/Offset																				
Counter length																				
Compiler																				
Replace with syn.																				

Retrieval Type	D				B				O				T							
													dummies=Y N				dummies=D P			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓				✓	✓			✓				✓	✓			✓			
Linked verification																				
Mark implementation						✓		✓						✓		✓				
No. abstract lines	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
Natural options																				
Owner	✓				✓	✓			✓				✓	✓			✓			
With users	✓				✓	✓			✓				✓	✓			✓			
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓			
Procedure code																				
Rules																				
Show implementation																				
Sorted by field																				
Subquery																				
Synonyms																				
STARTAB elements																				
Trigger																				
Use Con-form	✓				✓	✓			✓				✓	✓			✓			
User exit	✓				✓				✓				✓				✓			
3GL specification																				

Output Options for User Retrieval (Continued)

Retrieval Type	U				E				C				Y				X			
Output Mode	D		L		T		X		L		D		D		L		X			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																				
Adabas sizes																				
Association attributes					✓	✓														
Attributes	✓					✓		✓					✓					✓		
Check expression																				
Composed fields																				
Connecting character						✓		✓				✓						✓		
Cover page	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		
Description	✓							✓				✓	✓					✓		
Display length																				
Display modifier	✓												✓							
Dummy/Placeholder						✓		✓										✓		
DV-field expression																				
Entry points																				
Extract	✓					✓		✓			✓	✓	✓					✓		
Generation layout																				
Adabas version																				
Language																				
Alignment/sync.																				
Position/Offset																				
Counter length																				
Compiler																				
Replace with syn.																				

Retrieval Type	U				E				C				Y				X			
Output Mode	D		L		T		X		L		D		D		L		X			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓					✓		✓				✓	✓					✓		
Linked verification																				
Mark implementation						✓		✓										✓		
No. abstract lines	✓		✓			✓		✓		✓		✓	✓		✓			✓		
Natural options																				
Owner	✓					✓		✓				✓	✓					✓		
With users	✓											✓	✓					✓		
Page size	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓		✓		✓	✓		
Procedure code																				
Rules																				
Show implementation																				
Sorted by field																		✓		
Subquery																				
Synonyms																				
STARTAB elements																				
Trigger																				
Use Con-form	✓							✓				✓	✓					✓		
User exit	✓												✓							
3GL specification																				

Owner Maintenance

Linking Objects Logically using Owners

You can logically connect a user and another object in Predict by means of an owner, for example to document who uses an object or who is responsible for it. Enter an owner in the owner list of a user and the same owner in the owner list of the object.

The following rules apply when assigning owners:

- An owner is created by adding its ID to at least one owner list of a Predict object of type user.
- Any user can belong to several owners.
- The owner list of dictionary object can contain up to 99 owners.

Maintaining the Owner List of an Object

These lists can be edited using the Predict Link Editor. See Chapter **Editors in Predict** in the *Predict Reference Manual*. The editor is invoked in one of the following ways:

- By entering *Y* in the Field *EDIT Owner* in the bottom line of any *Add*, *Copy* or *Modify* screen.
- With function *Edit owners of an object* in a maintenance menu.
- With command `EDIT<object-type>OWNER<object ID>`.

Disallowing or Forcing Owner Entries

The data dictionary administrator can make the adding of owners optional, prohibited or mandatory by setting the metadata administration parameter *Edit owner* to *Allow*, *Disallow* or *Force*. This parameter can be specified for each object type. If the *Edit owner* parameter is set to *Allow*, any user can specify a default to be displayed in the *EDIT owner* parameter of *Add/Copy/Modify* screens.

Owner-specific Maintenance Functions

Rename/Merge Owner (Code R)

Owners can be renamed using the function *Rename / Merge owner*.
After the function has been performed, the old owner will no longer exist.

```
17:36:13          ***** P R E D I C T  4.1.1  *****          1999-02-24
                    - Rename/Merge Owner -

Owner ID ..... JCA

Enter new owner ID .. JCAX

                2 objects with this owner will be updated.

Enter '.' to return to menu.
```

If the owner name specified as the new owner already exists, the function assigns all objects of one owner to another owner. Additional confirmation is requested before this operation is carried out. (“*New owner ID already exists. Move the assigned objects from one owner to another owner ID. (Y/N)*”).

Example

The owner *Smith*, who is assigned to 24 Program objects, is renamed to the existing owner *Miller* (because Mr. Smith accepted another assignment).
Mr. Miller now has an additional 24 Programs assigned to him.

After the *Rename/Merge owner* function has been performed, the objects that have been updated are listed.

Purge Owner (Code *E*)

The following rules apply when purging owners.

- The function cannot be executed if an object has only this owner in its owner list and OWNER=FORCE has been defined for this object.
- If you confirm the function with DELETE, the owner is deleted from the owner list of all objects.
- The number of objects affected by the DELETE option is displayed before the owner is actually purged.

Owner Retrieval

```

17:37:34          ***** P R E D I C T  4.1.1  *****          1999-02-24
Plan   4          - (OW) Owner Retrieval -                      Profile JCA

Retrieval Type

D  Owners
O  Owners with no user
U  Objects with no owners
X  Cross reference owners

Retrieval type ....
Output mode .....* S Select

Owner ID ..... JCA

Output options ..*   Profile JCA                               Related type....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next

```

Note:

Owners cannot be used as additional selection criteria (restrictions) when retrieving information on owners.

Owner-specific Retrieval Functions

Owners with no User (Code *O*)

Lists owners which are not assigned to any user.

Direct command: FREE OWNER

Valid output modes: *List, Select.*

Objects with no Owners (Code U)

Reports on objects that have no owner.
 Command: **EMPTY OWNER**
 Valid output modes: *List, Select.*

Note:
 It is not possible to select objects for immediate processing from lists produced with the output mode Select. Objects can however be selected for later processing from the workplan.

Cross Reference Owners (Code X)

Lists all objects, that have specified owners in their owner list.
 Command: **XREF OWNER**
 Valid output mode: *Cross reference.*

```

17:39:12          ***** P R E D I C T  4.1.1  *****          1999-02-24
                    - Cross Reference for Owner -

Owner ID ..... BOE
-----
Program ID ..... C-PR-P
Keywords
COO
Extracts
  HEB-TEST, STK-ET-2, ARH-ET-0, BOE-ALL
Owner ID
  HEB
    ? User ID   User name
    ?   HEB-1
    ?   HEB-PUR   TEST
  GER
    ? User ID   User name
    ?                               >>> No users exists <<<
  BOE
    ? User ID   User name

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
          Quit          RFind Flip  -      +          Left  Right
  
```

Layout of Owner Lists

17:40:07	***** P R E D I C T 4.1.1 *****	1999-02-24
	- List Owner -	

Cmd	Owner ID	User ID User name
1	B	>>> No users exists <<<
2	BER	>>> No users exists <<<
3	BOE	BOE1
		BOE2
4	BOE-OW01	BOE3

Meaning of Columns

Owner ID	ID of owner.
User ID	ID of the person the owner represents.
User name	Name of the person the owner represents. *** <i>multiple</i> *** is displayed if an owner is assigned to more than one user. ? is displayed if an owner is not yet assigned to a user.

Layout of Owner Lists for Users

Meaning of Columns

Owner ID	ID of owner.
Others related User	Number of other users which are related to this owner.
Objects	Number of objects except users which are related to this owner.

Output Options for Owner Retrieval

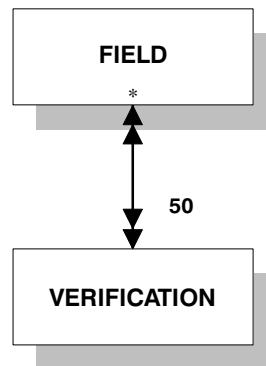
Retrieval Type	D				O				U				X			
Output Mode			L				L				L		X			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																
Adabas sizes																
Association attributes																
Attributes														✓		
Check expression																
Composed fields																
Connecting character														✓		
Cover page			✓				✓				✓		✓	✓		
Description														✓		
Display length																
Display modifier																
Dummy/Placeholder														✓		
DV-field expression																
Entry points																
Extract														✓		
Generation layout																
Adabas version																
Language																
Alignment/sync.																
Position/Offset																
Counter length																
Compiler																
Replace with syn.																

Retrieval Type	D				O				U				X			
Output Mode			L				L				L		X			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords													✓			
Linked verification																
Mark implementation											✓		✓			
No. abstract lines											✓		✓			
Natural options																
Owner													✓			
With users													✓			
Page size (<i>only in batch or printout</i>)			✓				✓				✓	✓	✓	✓		
Procedure code																
Rules																
Show implementation																
Sorted by field											✓		✓			
Subquery																
Synonyms																
STARTAB elements																
Trigger																
Use Con-form													✓			
User exit																
3GL specification																

VERIFICATION

Objects of type *verification* can contain code for processing rules. Verifications can have as status *documented*, *conceptual*, *free*, *automatic*, *Natural Construct* or *SQL*.

In the Predict metastructure, verifications can have parents of type *field* or *user-defined*.



How this Chapter is Organized

- **Verification Maintenance** page 428
 - The *Verification Maintenance* Menu, page 428
 - The *Add a Verification* Screen, page 430
 - Function *Purge Verification*, page 434
 - Rename/Change Status of a Verification, page 434
 - Edit Rule of a Verification, page 434
- **Verification Retrieval**, page 437
 - Verification-specific Retrieval Parameters, page 437
 - Verification-specific Retrieval Functions, page 437
 - List Verifications to Regenerate, page 437
 - Layout of Verification Lists, page 438
 - Output Options, page 439

Additional Information on Verifications/Processing Rules

- See Chapter **Verifications and Processing Rules** in the Manual *Predict and Other Systems*.
- The editor used to modify processing rules is described in Chapter **Editors in Predict** in the *Predict Reference Manual*.
- See also **Rippling Verifications** in Chapter **File** in the Manual *Predefined Object Types in Predict*, page 271.

The Verification Maintenance Menu

The *Verification Maintenance* menu is called with function code *M* and object code *VE* in a main menu or with the command MAINTAIN VERIFICATION.

```

09:28:30          ***** P R E D I C T  4.1.1  *****          1999-02-13
Plan    3          - (VE) Verification Maintenance -          Profile JCA

Function                                Function
A  Add a Verification                    D  Display Verification
C  Copy Verification                    L  Link children
M  Modify Verification                  O  Edit owners of a Verification
N  Rename/change status verific.        R  Edit rule of a Verification
P  Purge Verification                  S  Select Verification from a list
W  Edit description

Function .....
Verification ID ..                      Status .....*
Copy ID .....                          Format .....*

Restrictions ....*   Profile JCA ,used   Child type .....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12---
      Help  Menu  Canc  S-fi  E-el  M-pr  Print Impl  Last  FLIP  PROF  Next

```

Parameters

Function	<p>Select a code from the menu to execute one of the maintenance functions.</p> <p>Standard maintenance functions are described in Chapter Maintenance in the <i>Predict Reference Manual</i>.</p> <p>The functions <i>Purge Verification</i>, <i>Rename/Change Status of a Verification</i> and <i>Edit Rule of a Verification</i> (Code <i>R</i>) are described on page 434.</p>
Verification ID	<p>Identifier of the Predict verification object.</p> <p>The identifier of a verification is checked against Natural naming conventions.</p>

Parameters

Copy ID	For <i>Copy</i> function: ID of new verification to be created.
Status	Status of the verification: A Automatic C Conceptual D Documented (no rule) F Free N Natural Construct S SQL <i>blank</i> any For the <i>Select</i> function: a status can be specified as an additional selection criterion.
Format	Format of the verification: A Alphanumeric B Binary D Date/time K Function key L Logical N Numeric <i>blank</i> Unknown (no rule defined) For the <i>Select</i> function: a format can be specified as an additional selection criterion.
Restrictions	Additional criteria can be specified to restrict the scope of dataspace to be processed. See Restrictions in Chapter The User Interface in the Manual <i>Introduction to Predict</i> .
Child type	For function <i>Link children</i> : objects of this type are to be linked to the dataspace. Valid values: <i>user-defined</i> .

The Add a Verification Screen

The screen is displayed for the *Add a Verification* function. The *Copy* and *Modify* screens are similar

09:27:57

***** P R E D I C T 4.1.1 *****

1999-02-13

- Modify Verification -

Verification ID . TEST-TOWN

Modified 1999-07-07 at 11:46

Status Free

by HO

Keys ..

Zoom: N

Format* A Alphanumeric

Modifier Zoom: N

Type* T Table of values

Message nr

Replacement 1 ...

Replacement 2 ...

Replacement 3 ...

Message text No SAG-office in that town.

Abstract Zoom: N

Values Zoom: N

BRUESSEL

RESTON

PARIS

DERBY

CAMBRIDGE

DARMSTADT

EDIT: Owner: N Desc: N * Rule: N

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Parameters

Verification ID	The identifier of the verification.
Status	The status assigned by Predict to the verification rule. See page 429 for list of valid values.
Format	The format of the verification rule. See page 429 for list of valid values.

Parameters

Modifier	<p>User and or user groups defined in Natural Security who can be authorized to modify free rules of the verification. The parameter is evaluated by Predict according to the setting of the default parameters <i>Rule in Map Editor / Rule in SYSDIC</i>. If any of these parameters is set to force, Predict checks the following:</p> <ul style="list-style-type: none">- that at least one modifier is specified,- that each modifier of the object is a Natural Security administrator, person or group,- that the user is listed as a modifier of the object. <p>See also description of <i>Rule in Map Editor / Rule in SYSDIC</i> in the Chapter Defaults in the <i>Predict Administration Manual</i> and Protecting Processing Rules in Chapter Protecting External Objects in Predict with Natural Security in the <i>Predict Security Manual</i>.</p>
Type	<p>The type of rule. Enter single-character code as shown in the following table. The table also shows the number of values to be specified with each type of rule and the generated code:</p>

Parameters

Code	Type of Rule	No. of Values	Generated Natural Statements	Generated SQL Clause
E	Equal to	<i>n</i>	IF NOT (&= value)	& = value1
		0 or 1	IF NOT & /* for format <i>logical</i>	& = value2 ...
G	Greater than	1	IF & LE value	& => value1
L	Less than	1	IF & GE value	& =< value1
N	Not equal to	<i>n</i>	IF (&= value1 OR= value2 ...)	& ^= value 1
		0 or 1	IF & /* for format <i>logical</i>	& ^= value 2 ...
R	Range of values	2	IF NOT (&= value1 THRU value2)	\$ between value1 and value2
T	Table of values	<i>n</i>	IF NOT (&= value1 OR= Value2 ...)	& in (value1, value2...)
U	User routine			
B	Range, but not	3	IF NOT (&= value1 THRU value2 BUT NOT value3)	& between value1 and value2 and & ^=value3
		4	IF NOT (&= value1 THRU value2 BUT NOT Value3 THRU value4)	& between value1 and value2 and & not between value3 and value4
I	Not in range	2	IF (&= value1 THRU value2)	¬ between value1 and value2
<i>blank</i>	(none) – no rule defined			

See also **Edit Rule of a Verification** on page 434.

Message nr	Number of Natural error message. The message will be displayed if a validation fails. Up to three replacement strings can be inserted into an error message if the respective targets (:1:, :2:, :3:) are provided.
Replacement 1 - 3	Strings to be inserted into a Natural message. See description of <i>Message nr</i> above.
Message text	Message to be displayed if a validation fails. A standard message will be created if neither <i>Message text</i> nor <i>Message nr</i> have been specified.

Parameters

Values

The values used to perform the verification. The following rules apply:

- The number of values to be specified depends on the verification type. See table above.
- Values are delimited
 - with blanks
 - with the Natural INPUT delimiter character (ID) defined in the Natural environment
 - by entering them in separate lines.
- Hexadecimal values can be specified in two ways:
 - if *Format=B*, hexadecimal values can be specified directly.
Example: *F0*
 - if *Format=A*, hexadecimal values must be preceded by uppercase *X* or *H* and be enclosed in single quotes.
Example: *X'F0'* or *H'F0'*
- Blanks can be specified in one of the following ways:
' ', *BLANK* or *SPACE*.
Strings that include blanks must be enclosed in single quotes, apostrophes in strings have to be doubled (for example: *'six o'clock'*).
- Line comments can be specified by preceding them with */** (a slash and an asterisk). Line comments can be used by SYSHELP as descriptive text in input windows.
Strings that include the comment delimiter */** must be enclosed in single quotes.

Verification-Specific Maintenance

Purge Verification (Code *P*)

A verification of type *automatic* cannot be purged. To purge a verification of this type, perform the following steps:

- Remove all links from fields to the verification
- Regenerate DDMs that were generated from the files linked to these fields.

When the verification is no longer connected to any fields, the status is changed to conceptual and the rule can be purged.

Rename/Change Status of a Verification (Code *N*)

The following rules apply when renaming a verification or changing its status:

- The name/status of the verification is changed in all verifications lists of fields.
- The status can only be changed from *Conceptual* to *Free* and vice versa.

Edit Rule of a Verification (Code *R*)

Processing rules of verifications are edited with the Predict Verification Editor. This editor can be invoked in one of the following ways:

- By entering *Y* in the field *EDIT Rule* in the bottom line of the *Add a Verification*, *Copy Verification* or *Modify Verification* screen.
- By calling the function *Edit rule* in the *Verification Maintenance* menu (Code *R*).
- By entering the direct command EDIT VERIFICATION RULE <*Verification-ID*>

Note:

Statements of the rule must not contain statement references to line numbers; use labels instead.

The Rule Editor

This section describes rule-specific editor commands. General editor commands are described in Chapter **Editors in Predict** in the *Predict Reference Manual*. See also Chapter **Verifications and Processing Rules** in the Manual *Predict and Other Systems*.

Editor Commands

CAT [[FREE] RET[URN]], SA[VE] [[FREE] RET[URN]]	Catalog/save the edited rule as a free rule. This command is only available when creating new rules and when editing conceptual rules.
<i>Note:</i>	Note that the SAVE or CAT command do not perform a syntax check. The syntax is checked however, when cataloging a map that uses a rule.
C[HECK]	Check whether the edited rule’s Natural syntax is valid and report errors.
GEN[ERATE]	Generate a rule from the values defined in the verification and add it to the end of the Natural source in the rule editor. This command is not available for verifications of type <i>U</i> . The table below shows which Natural statements are generated for the different types of verifications:
GEN[ERATE] N	Generates a rule for Natural CONSTRUCT from a verification of status <i>documented (D)</i> . The status of the verification will be changed to <i>N</i> .
GEN[ERATE] S	Generates a rule for Adabas SQL Server from a verification of status <i>documented (D)</i> . The status of the verification will be changed to <i>N</i> .
GLOBALS SM=OFF	Switch to the reporting mode of Natural.
GLOBALS SM=ON	Switch to the structured mode of Natural.
RENUM[BER], N	Renumber the source lines in steps of 10 and renumber references to them accordingly.

RUN, CHECK

Check the edited rule. If no errors are found, a map is produced with which the user can test the rule by entering input values. The following rules apply:

- Length and format of the input field are derived from the rule format. Whereas CHECK derives the format without further notification, RUN displays an additional window where the derived field length is displayed and can be overwritten.

Rule Format	Format of the derived field	Length of the derived field
A	A	66
B	B	33
D	D	
L	L	1
N	N	27

- *RUN* tests a rule of format *K* (function key) without input data.
- For a rule of format *L* (logical), a blank space means *false* and any other input value means *true*.
- The stack must not be changed.
- The contents of the source area must not be changed.

Note: All variables used except the ampersand (&) must be defined within the code.

- The variable names *SYSDIC-C1* and *SYSDIC-C2* are used for internal purposes and must not be used within the rule.
- The source will be renumbered.

Verification-Specific Retrieval

Verification-Specific Retrieval Parameters

verif. of status	Limits the scope of the function to verifications with the status specified. Valid values: A Automatic C Conceptual D Documented (no rule) F Free S SQL N Natural Construct
format	Limits the scope of the function to verifications with the format specified. Valid values: A Alphanumeric B Binary D Date/time K Function key L Logical N Numeric

Verification Specific Retrieval Functions

List Verifications to Regenerate (Code K)

Lists verifications whose definitions have been modified since a DDM was generated containing a field that uses one of the verifications.

Direct command: REGENERATE VERIFICATION.

Layout of Verification Lists

12:15:19	***** P R E D I C T 4.1.1 *****	1999-02-17
	- List Verification -	

Cnt	Verification ID	S F Comp. F T
1	JCA-PR1	D U
2	JCA-S	D U
3	* JCA-VE1	A A B E
	Verification values	
	Samstag	
4	JCA-VE2	D A U
5	JP-TEST	F A B E
	Verification values	
9		

Meaning of Columns

Verification ID	ID of the Predict verification object.
S	The status of the verification rule. See page 429 for list of codes and values.
F	The format of the verification rule. See page 429 for list of codes and values.
Comp. F	Compatible format. Not all formats are compatible with all verification types.
T	Type of the verification. See table on page 432.
Values	Verification values.

Output Options for Verification Retrieval

Retrieval Type	D				B				O				T							
													dummies=Y N				dummies=D P			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																				
Adabas sizes																				
Association attributes					✓	✓	✓	✓					✓	✓	✓	✓				
Attributes	✓				✓				✓				✓				✓			
Check expression																				
Composed fields																				
Connecting character						✓								✓						
Cover page	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Description	✓				✓	✓			✓				✓	✓			✓			
Display length																				
Display modifier	✓				✓				✓				✓				✓			
Dummy/Placeholder														✓		✓		✓		✓
DV-field expression																				
Entry points																				
Extract	✓				✓	✓			✓				✓	✓			✓	✓		
Generation layout																				
Adabas version																				
Language																				
Alignment/sync.																				
Position/Offset																				
Counter length																				
Compiler																				
Replace with syn.																				

Retrieval Type	D				B				O				T							
													dummies=Y N				dummies=D P			
Output Mode	D		L		D		L		D		L		D		L		D		L	
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓				✓	✓			✓				✓	✓			✓			
Linked Verification																				
Mark implementation	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
No. abstract lines	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓		✓	
Natural options																				
Owner	✓				✓	✓			✓				✓	✓			✓			
With users	✓				✓	✓			✓				✓	✓			✓			
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓
Procedure code																				
Rules	✓				✓				✓				✓				✓			
Show implementation	✓				✓				✓				✓				✓			
Sorted by field						✓		✓												
Subquery																				
Synonyms																				
STARTAB elements																				
Trigger																				
Use Con-form	✓				✓	✓			✓				✓	✓			✓			
User exit	✓				✓				✓				✓				✓			
3GL specification																				

Output Options for Verification Retrieval

Retrieval Type	U				E				C				K			
Output Mode	D		L		T		X		L		D		L			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Adabas attributes																
Adabas sizes																
Association attributes					✓	✓										
Attributes	✓					✓		✓								
Check expression																
Composed fields																
Connecting character						✓		✓				✓				
Cover page	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Description	✓							✓				✓				
Display length																
Display modifier	✓															
Dummy/Placeholder						✓		✓	✓		✓					
DV-field expression																
Entry points																
Extract	✓					✓		✓			✓	✓				
Generation layout																
Adabas version																
Language																
Alignment/sync.																
Position/Offset																
Counter length																
Compiler																
Replace with syn.																

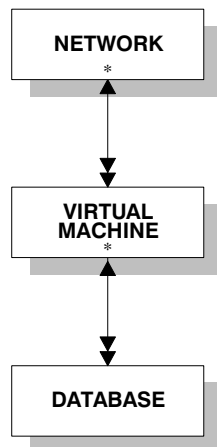
Retrieval Type	U				E				C				K			
Output Mode	D		L		T		X		L		D		L			
Current/Related	c	r	c	r	c	r	c	r	c	r	c	r	c	r	c	r
Keywords	✓					✓		✓				✓				
Linked Verification																
Mark implementation	✓		✓		✓	✓	✓	✓		✓			✓			
No. abstract lines	✓		✓			✓		✓		✓		✓	✓			
Natural options																
Owner	✓					✓		✓				✓				
With users	✓											✓				
Page size (<i>only in batch or printout</i>)	✓		✓		✓	✓	✓	✓	✓	✓	✓	✓	✓			
Procedure code																
Rules	✓															
Show implementation	✓															
Sorted by field																
Subquery																
Synonyms																
STARTAB elements																
Trigger																
Use Con-form	✓							✓				✓				
User exit	✓															
3GL specification																

VIRTUAL MACHINE

Since data can be distributed across several databases, the exact location of data storage has to be specified: database are linked to objects of type *virtual machine* and virtual machines are linked to objects of type *network*.

The Predict object *virtual machine* identifies the hardware and operating system environment of a database.

See Chapter **Adabas Star** in the Manual *Predict and Other Systems* for a complete description of how to define distributed data structures with Predict.



In the predefined Predict metastructure, a *virtual machine* has to be related to a *network*, and *databases* can be related to the virtual machine.

Links between networks, virtual machines and databases are established with the parameters *in network* and *in virtual machine*, and not with child/parent associations.

How this Chapter is Organized

- **Virtual Machine Maintenance**
 - The *Virtual Machine Maintenance* Menu, page 445
 - The *Add a Virtual Machine* screen, page 446
- **Virtual Machine Retrieval** page 447
 - Virtual Machine Specific Retrieval Parameters, page 447
 - Layout of Virtual Machine Lists, page 447

The Virtual Machine Maintenance Menu

The *Virtual Machine Maintenance* menu is called with function code *M* and object code *VM* in a Predict main menu or the command MAINTAIN VIRTUALMACHINE.

```
16:22:38          ***** P R E D I C T  4.1.1  *****          1999-03-03
Plan    0          - (VM) Virtual machine Maintenance -          Profile JCA

Function                                Function

A  Add a Virtual machine                D  Display Virtual machine
C  Copy Virtual machine                 L  Link children
M  Modify Virtual machine                O  Edit owners of a Virtual machine
N  Rename Virtual machine                S  Select Virtual machine from list
P  Purge Virtual machine                 W  Edit description

Function .....
Virtual machine ID ....
Copy ID .....
In Network .....

Restrictions .....*   Profile JCA ,used           Child type ....*

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help Menu Canc S-fi E-el M-pr Print Impl Last FLIP PROF Next
```

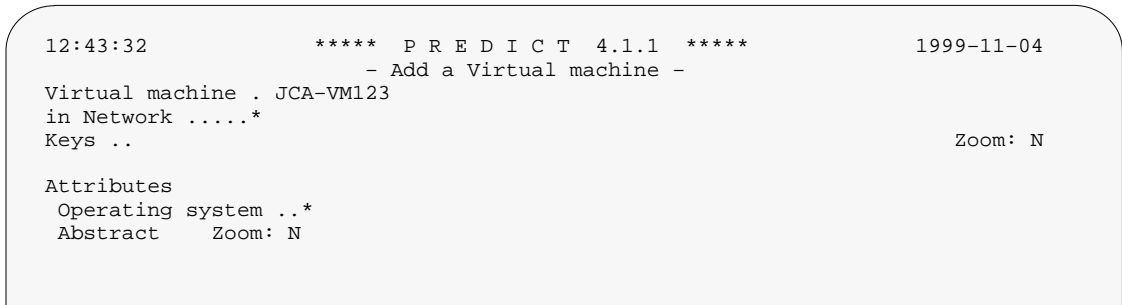
Parameters

Note:
Parameters not listed here are described under **Global Attributes**, page 6.

Function	Standard maintenance functions are described in Chapter Maintenance in the <i>Predict Reference Manual</i> .
In Network	ID of the network containing the virtual machine.

The Add a Virtual Machine Screen

The screen is displayed for the *Add a Virtual Machine* function. The *Copy* and *Modify* screens are similar.



Parameters

Note:

Parameters not listed here are described under **Global Attributes**, page 6.

- Virtual machine The ID of the virtual machine.
- in Network The ID of the network containing the virtual machine.
- Operating system Valid values:

BS2000/OSD	FACOM/XA	UNIX	undefined
BS2/XS	HP-UX	VMS	
CMS	OS/390	VSE/XA	
CMS/XA	MVS/ESA	VS/XA	
DOS	MVS/XA	WANG/VS	
DOS/VS	OS	WNT	
DOS/VSE	OS/2	WNT-AXP	
FACOM	SINIX	WNT-X86	

Virtual Machine Retrieval

Virtual Machine Specific Retrieval Parameters

in Network

Only virtual machines related to the network will be included in the report.

Layout of Virtual Machine Lists

11:13:46	***** P R E D I C T 4.1.1 *****	1999-02-25
	- List Virtual machine -	

Cnt	Virtual machine ID	Operating system
1	ARH-VM2	
2	ARH-VM4	MVS
3	BOE-TEST-1	
4	BOE-VM	
5	BOE-VM-CMS	CMS
6	BOE-VM-01	
7	BOE-VM01	MVS/XA

Meaning of Columns

Operating system

Operating system type of the virtual machine. See list on page 446.

Output Options for Virtual Machine Retrieval

The output options valid for this object type are identical to those for object type *dataspace*. See page 58.

INDEX

Numbers

3GL specification, Field attributes, 141

A

A/D, Field attribute (DB2), 137

Abstract

- line editor commands, 11

- parameter when redocumenting Programs, 363

- rippling, 278

Access level, file attribute (Adabas), 195

ACTIVE, editing child list of a Program, 356

ADA, edit field list command, 258

Adabas C, specifying the size of database, 21

Adabas C SQL usage, File attribute, 190

Adabas D, naming conventions for Adabas D tables and views, 217

Adabas D table, documenting, 218

Adabas D view, documenting, 219

Adabas security, database attribute, 20

Adabas short name. *See* Field short name

Adabas SQL Server

- documenting Adabas SQL objects, 212

- documenting Adabas SQL views, 216

- naming conventions for Adabas SQL Server tables and views, 212

Adabas SQL usage. *See* Adabas C SQL usage

Adabas Star, changing ADASTAR access only flag, 36

ADAMOFLOW, documenting, 198

ADAMPARM, documenting, 198

ADASTAR access number, file attribute, 190

ADASTAR access only, database attribute (Adabas), 20

ADASTAR number

- listing use of ADASTAR numbers, 326

- network retrieval function, 326

ADASTAR parameter

- changing, 35

- database attribute, 18

Add

- Dataspace, 50

- Field, 157

Alternate RABN, database attribute (Adabas), 22

ASSO

- documenting Adabas C dataset, 21

- padding factor, file attribute (Adabas), 193

Audit, File attribute, DB2 table, 222

Automatic allocation, file attribute (Adabas), 195

Automatic null value, profile parameter, 119

Average, File Relation attribute, 295

Average count, File attribute (Natural Construct), 185

B

Blksize, file attribute (ISAM), 252

Browse through Fields of a File, 164

Buffer pool

- database attribute (DB2), 24

- dataspace attribute, 51

- Field attribute (DB2), 151

Build/Extend an Extract, Extract maintenance function, 81

C

Cardinality, File Relation attribute, 294
 CAT, Rule Editor command, 436
 CCSID
 database attribute (DB2), 25
 dataspace attribute (DB2), 51
 file attribute, DB2 table, 222
 Character set, Field attribute (SQL), 102
 CHECK, Rule Editor command, 436, 437
 Check against standard
 Field attribute, 120
 Field maintenance parameter, uncoupling
 Standard Fields, 273
 rippling parameter, 268
 Check constraint name
 File attribute
 Adabas D table, 218
 General SQL File, 241
 ORACLE table, 235
 file attribute, DB2 table, 222
 Checkpoint file, database attribute (Adabas), 20
 CI size data/index, file attribute (VSAM), 248
 Ciphred, file attribute (Adabas), 195
 Close option
 dataspace attribute, 51
 Field attribute (DB2), 151
 Cluster column, File attribute, ORACLE table, 234
 Cluster index, Field attribute (DB2), 150
 Cluster name, File attribute, ORACLE table, 234
 COBOL, mapping of data structures in redefinitions, 162
 Collection name, packagelist attribute, 336
 Comment Lines, Field list of File, 260
 Comments, Global attribute. *See* Abstract
 Compressed file, file attribute (VSAM), 248
 Conceptual database, documenting, 23
 Conceptual files, documenting, 200

Condition name, Field attribute (3GL), 143
 Condition value, Field attribute (3GL), 144
 Constraint name, additional Field attribute, 155
 Contiguous, file attribute (ISAM), 252
 Converting, upper/lower case, 7
 Copy
 Dataspace, 50
 Field, 157
 Copy Extracts, Extract Maintenance function, 67
 Copy ID, global attribute, 9
 Cross reference
 Keyword, 309
 Owners, 421
 Cross reference fields, Field retrieval fuction, 170
 Cylinder, database attribute (Adabas), 22
 Cylinders/Blocks, file attribute (Adabas), 193

D

D, descriptor type, list of, 114
 DATA, documenting Adabas C dataset, 21
 Data capture, file attribute, DB2 table, 222
 DATA padding factor, file attribute (Adabas), 193
 Data sharing group member, database attribute (DB2), 25
 Database
 changing database number, 36
 changing ID, 33
 changing type, 34
 description of predefined type, 13
 layout of database lists, 41
 linking to another VM, 34
 purging, 31
 rename/renumber/retype, 33
 Database Maintenance Menu, 15
 Database name, File-Attribute, INGRES Table, 238

Predefined Objects in Predict

Dataspace
 definition of partitioned dataspace, 55
 delete, 57
 description of predefined type, 47
 layout of dataspace lists, 58

DB, Field short name, Field attribute, 117

DB2
 defining, DB2 index, 150
 documenting, database, 24
 documenting tables and views, 220

DB2 name, database attribute (DB2), 24

DB2 or SQL/DS, database attribute (DB2), 24

DB2 plan name, system attribute, 400

DB2 ROSHARE parm, database attribute (DB2), 25

DB2 table, documenting, 221

DB2 tables and views, naming conventions, 220

DB2 view, documenting, 223

DD Name, Field attribute (VSAM), 139

Default name, additional field attribute
 DB2, 154
 SYBASE, 154

Default owner, parameter when redocumenting programs, 364

Default storagespace, database attribute (DB2), 24

Delete
 Dataspace, 57
 Extract, 84
 Field, 158

Delete type, File Relation attribute, 295

Depending on, Field attribute (3GL), 142

Derived field expression, Field attribute (SQL), 149

Derived Fields
 defining, 123
 validation of, 133

Description, parameter when redocumenting programs, 363

Descriptor Type, attribute of Standard Field, rippling, 271

Descriptor type
 Field attribute, 114
 list of, 114

Device
 database attribute (Adabas), 22
 file attribute (Adabas), 193
 file attribute (ISAM), 252

Device and size specification, file attributes (Adabas), 193

Device type, storagespace attribute, 389

Difference of Files, file retrieval function, 281

Dir-blocks, file attribute (ISAM), 252

Distribution key, file attribute (Adabas), 197

DS reuse, file attribute (Adabas), 194

DSSIZE, documenting, 193

Duplicated, File attribute, INGRES table, 230

DV field expression, Field attribute (SQL). *See* Derived field expression

E

EDIT
 ASSO, database attribute (Adabas), 22
 DATA, database attribute (Adabas), 22

Edit
 Field expression, 164
 File list of Database, 38
 procedure code of a program, maintenance function, 366

Edit mask
 Field attribute, 121
 Field attribute (Adabas), 147

Edit program, File attribute, DB2 table, 222

Edit Rule of a Verification, verifications maintenance function, 435

EDIT subquery
 File attribute
 for SQL tables, 205
 for SQL views, 204
 file maintenance function, 209

 Edit Subquery of a File, file maintenance
 function (DB2), 264
 Edit/Link Objects, Extract maintenance
 function, 72
 End, Field attribute, defining derived Fields, 127
 Entire SQL usage. *See* Adabas C SQL usage
 ENTIRE SYSTEM SERVER, documenting
 ENTIRE SYSTEM SERVER files and
 userviews, 253
 Entire System Server, documenting Entire
 System Server nodes, 23
 Erase, file attribute (Adabas/UNIX), 196
 Erase opt, field attribute (DB2), 152
 Erase option, dataspace attribute, 54
 Explode IMS Databases, database retrieval
 function, 40
 Export, Extract, 72
 Extent allocation, file attributes (Adabas), 196
 External name
 file attribute, 181
 file attribute (ISAM), 251

 Extract
 description of object type, 63
 exporting, 72
 object editor, 72
 purge, 84
 search unused Extracts, 85

 Extract Maintenance, Menu, 65
 Extracts
 related to no object, retrieval function, 85
 related to objects, retrieval function, 85

F

F
 Field attribute, for derived Fields, 127
 Field formats, list of, 101

 FC, Field attribute (3GL), 144
 Field Format, list of, 101
 Field ID, file attribute (ADAM), 198
 Field ID/Synonym, Field-specific retrieval
 parameter, 165
 Field length, adjusting when generating data
 definitions, 103
 Field Name Synonyms, Field attribute (3GL),
 145
 Field short name, Field attribute, 117
 Field types, list of, 99
 Fields, 91
 changing Field format, 129
 layout of Field lists, 171
 list of formats, 101
 retrieval parameters, 165
 sub/superfields/descriptors, determining
 format, 129

 Fields Related to a Z-file, Field retrieval
 function, 169
 Fields with no Verification, Field retrieval
 function, 170
 Fields with Verification, Field retrieval function,
 170
 Figurative constant, Field attribute (3GL), 144
 File, valid file no. for various file types, 185
 File criteria, file attribute (Adabas), 197
 File for select, profile parameter, 273
 File ID, file attribute, 183
 File list, parameter when redocumenting
 programs, 364
 File Maintenance Menu, 180
 File number, file attribute, 182, 185

Predefined Objects in Predict

File Relation

- checking for validity, 296
- layout of File Relation list, 298
- predefined object type, 291
- specific retrieval parameters, 298

Files

- change Adabas C SQL usage, 257
- change file number, 257
- change file type, 257
- change logical ADASTAR type, 257
- change master file, 257
- creating Standard Files, 269
- description of predefined object type, 177
- layout of file lists, 284
- list of valid file types, 181
- valid database types for different file types, 184
- valid master files for different file types, 186

Files of type, file attribute, 181

Files Related to a File, file retrieval function, 283

FILLER, use in Field redefinitions, 160

First default keyword, parameter when redocumenting programs, 364

First owner ID, user attribute, 409

FLIP C, editor command, 258

FLIP T, editor command, 258

Force Standard, file maintenance function, 261

Format, verification attribute, 430

Free pages

- datspace attribute, 53
- Field attribute (DB2), 152

Free space, file attribute (VSAM), 248

G

GBPCACHE, dataspace attribute (DB2), 54, 152

GEN[ERATE] N, Rule Editor command, 436

GEN[ERATE] S, Rule Editor command, 436

General SQL files, documenting, 200

General SQL server, documenting , 23

GENERATE, Rule Editor command, 436

Global attributes, of predefined object types, 7

GLOBALS, Rule Editor command, 436

Gr.structur, Field attribute (3GL), 141

GROUP BY clause (SQL), documenting, 210

H

Handle /* in columns 1+2 as comment or remark, default parameter, function Redocument Program, 365

HAVING clause (SQL), documenting, 210

Hyperdescriptors, defining, 135

I

Implementation Pointer, program attribute, 348

Implode Fields, Field retrieval function, 170

IMS

documenting IMS database, 27

documenting segment layouts and userviews, 243

field lists of IMS files, 245

restrictions applying when editing a field list of an IMS file, 245

IMS name, database attribute (IMS), 27

IMS or DL1, database attribute (IMS), 27

IMS type, database attribute (IMS), 27

in database, file attribute, 182, 184

in file, Field-specific retrieval parameter, 166

in files of type, Field-specific retrieval parameter, 166

In network, virtual machine attribute, 446

Index name, Field attribute (DB2), 137, 150

Index on PE Group Level, Field attribute, 121

Index required, field attribute (DB2), 151

Indexed by, Field attribute (3GL), 142

INFORMIX, naming conventions for

INFORMIX objects, 224

INFORMIX Handler, database type, defining, 30

INFORMIX table, documenting, 225
 INFORMIX view, documenting, 227
 INGRES, naming conventions for INGRES
 objects, 228
 INGRES Handler, database type, defining, 30
 INGRES table, documenting, 229
 INGRES view, documenting, 231
 Init. value, Field attribute (3GL), 142
 Initial, File attribute, ORACLE table, 235
 Initialize with, Field attribute (3GL), 142
 Intrans, File attribute, ORACLE table, 234
 Interface, description of predefined object type,
 299
 ISAM, documenting files and sequential files,
 251
 ISN reuseage, file attribute (Adabas), 194
 ISN Size, file attribute (Adabas), 195

J

Journaling, File attribute, INGRES table, 230
 JUSTIFIED RIGHT, documenting COBOL, 141
 Justify, Field attribute (3GL), 141

K

Keyword, description of predefined object
 type, 303
 Keywords
 finding unused keywords, 309
 layout of Keyword lists, 309
 link/unlink, 306
 purging, 305

L

Language, program attribute, 344
 Large, dataspace attribute, 50
 Layout of Field lists, 171
 Length, Field attribute, 103
 for derived Fields, 127

 Level number, Field attribute, 100
 Lib.name, parameter when redocumenting
 programs, 363
 Library
 list of valid libraries, 346
 program attribute, 346
 system attribute, 400

 Library Structure, description of object type,
 315
 Library, DBnr, Fnr, parameter when
 redocumenting programs, 365
 Line comments, for Verification values, 434
 Link, Keyword to object, 306
 Link children, Library Structure maintenance
 function, 318
 Link objects to extract, Extract Maintenance
 function, 78
 Link to system, parameter when redocumenting
 programs, 362
 Link Verification, Field maintenance function,
 164
 List Keywords Related to no Object, keyword
 retrieval function, 309
 List Verifications to Regenerate, verification
 retrieval function, 438
 Literal name, File attribute (Natural Construct),
 185
 Load-Lib, program attribute, 348
 Location name, packagelist attribute, 337
 Lockmax, dataspace attribute (DB2), 51
 Lockpart, dataspace attribute (DB2), 51

Predefined Objects in Predict

Locksize, Dataspace attribute
DB2, 51
SQL/DS, 56

Logical ADASTAR type, file attribute, 187
LOWNERID, file attribute (Adabas), 195

M

Master file, file attribute, 186
Max ISN, file attribute (Adabas), 192
Max recl., file attribute (Adabas), 194
Max. length, file attribute (IMS), 244
Maxextents, File attribute, ORACLE table, 235
MAXFILES, ADADEF parameter, 20
Maximal files, database attribute (Adabas), 20
Maximum, File Relation attribute, 295
Maximum secondary allocation, file attribute (Adabas), 194
Maxrows, dataspace attribute (DB2), 51
Maxtrans, File attribute, ORACLE table, 234
Member, program attribute, 345
Member cluster, dataspace attribute (DB2), 51
Member type, program attribute when redocumenting programs, 360
Message nr, verification attribute, 433
Message text, verification attribute, 433
Method, description of predefined object type, 319
Min ISN, file attribute (Adabas), 192
Min. length, file attribute (IMS), 244
Minextents, File attribute, ORACLE table, 235
Minimum, File Relation attribute, 294
Mirror, file attribute (Adabas), 195
Mode, program attribute, 348
Modifier, verification attribute, 432
Modify, Dataspace, 50
Modify Adabas attributes, file maintenance function, 264
Modify STARTAB Elements, file maintenance function, 264

MORE Size, database attribute (Adabas), 20
Move Field within File, 158
Multiple-master view, coupling with Master File, rippling, 276

N

Naming conventions
for Adabas D tables and views, 217
for Adabas SQL Server tables and views, 212
for DB2 tables and views, 220
for Extracts, 8
for INFORMIX objects, 224
for INGRES objects, 228
for objects used by Natural, 8
for ORACLE objects, 232
for Predict objects, 7
for SQL objects, overview, 202
for standard files, 8
for SYBASE objects, 237
NAT Security (FSEC), database attribute (Adabas), 20
NAT-Func, program attribute, 348
NAT-I, Natural field length, Field attribute, 120
Natural, mapping of data structures in redefinitions, 161
Natural Construct parameters, File attributes, 185
Natural Construct parameters for Files
Average count, 185
Literal name, 185
Stability, 185
Natural Header, Field attribute, 121
Natural statements, generated by rule editor, 433
Natural VSAM handler, documenting VSAM clusters, 29
Networks
description of predefined object type, 323
purging, 325
Next, File attribute, ORACLE table, 235
NISIZE, documenting, 193

Node, description of object type, 329
 Non-Standard Fields, Field retrieval function, 169
 Nr of partitions, dataspace attribute, 50
 NULL, edit field list command, 259
 Null Default Option, Field attribute, SQL Files, 119
 Null Flag, Field attribute (VSAM), 139
 Number of partitions
 Field attribute (DB2), 150
 File attribute, DB2 table, 222
 Number of RABN, database attribute (Adabas), 22
 Numeric zones, file attribute (VSAM), 248

O

OBid, File attribute, DB2 table, 222
 Object ID, global attribute, 7
 Objects, linking logically, using Owner, 417
 Objects with no Owners, owner retrieval function, 421
 Occ, field attribute, 116
 Adabas SQL server, 116
 One AC extent, file attribute (Adabas), 194
 Operate on Extracts, Extract maintenance function, 67
 Operating system, virtual machine attribute, 447
 ORACLE, naming conventions for ORACLE objects, 232
 ORACLE Handler, database type, defining, 30
 ORACLE table, documenting, 233
 ORACLE view, documenting, 236
 Org, file attribute (VSAM), 250
 Organization, file attribute (ISAM), 252
 Other file types, documenting, 200
 Other Handler, documenting, 26
 Output Options, for Owner retrieval, 424

Output options
 for Database retrieval, 42
 for Dataspace retrieval, 59
 for Extract retrieval, 86
 for Field retrieval, 173
 for File Relation retrieval. *See* Output options for Dataspace retrieval
 for File retrieval, 286
 for Keyword retrieval, 310
 for Library Structure retrieval. *See* Output options for Dataspace retrieval
 for Network retrieval. *See* Output options for Dataspace retrieval
 for Node retrieval. *See* Output options for Dataspace retrieval
 for Packagelist retrieval. *See* Output options for Dataspace retrieval
 for Program retrieval, 369
 for Report Listing retrieval, 381
 for Server retrieval. *See* Output options for Dataspace retrieval
 for Storagespace retrieval, 392
 for System retrieval. *See* Output options for Dataspace retrieval
 for User retrieval, 413
 for Verification retrieval, 440
 for Virtual Machine retrieval. *See* Output options for Dataspace Retrieval

Overflow, file attribute (ADAM), 198
 Owner, linking objects logically, 417
 Owners
 disallowing or forcing owner entries, 417
 layout of owner lists, 422
 renaming/moving, 418

Owners with no User, owner retrieval function, 420

Predefined Objects in Predict

P

Packagelists

- description of predefined object type, 333
- Layout of Packagelist Lists, 339
- purging, 338

Pages per segment, dataspace attribute, 52

Parent, file attribute (IMS), 244

Parm, file attribute (ADAM), 198

Password required

- dataspace attribute, 52
- Field attribute (DB2), 151
- storagespace attribute, 389

Pctfree, File attribute, ORACLE table, 234

Pctincrease, File attribute, ORACLE table, 235

Pctused, File attribute, ORACLE table, 234

Percentage for indices, dataspace attribute, 56

Percentage free

Dataspace attribute

DB2, 53

SQL/DS, 56

Field attribute (DB2), 152

Periodic Groups in Periodic Groups, defining, 100

Phonetic Descriptors, defining, File types A, C and Z, 134

Phys. file number, file attribute (Adabas), 192

Physical ADASTAR type, file attribute (Adabas), 192

Physical database number, database attribute, 18

Physical DBnr, database attribute (RMS/other handler), 26

Piece size, Field attribute (DB2), 151

PLOG, file attribute (Adabas), 195

Predict (FDIC), database attribute (Adabas), 20

Primary alloc, Field attribute (DB2), 151

Primary allocation, dataspace attribute, 54

Private dataspace, dataspace attribute, 56

Procedure name, Field attribute (DB2), 148

Procedure parameter, Field attribute (DB2), 148

Process, file attribute Entire System Server, 254

Processing option, parameter when

redocumenting programs, 362

Program, redocument, possible languages, 359

Program ID prefix, parameter when

redocumenting programs, 363

Program list, parameter when redocumenting programs, 364

Program of type, program attribute, 344

Programs

description of predefined object type, 341

documenting object modules, 352

editing child lists, 353

layout of program lists, 368

list of valid languages, 344

list of valid program types, 344

overview of language-specific program types, 354

purging, 358

redocumenting, 358

specific retrieval parameters, 367

system programs, 352

Property, description of predefined object type, 373

Purge

database, 31

Dataspace, 57

Extract, 84

Field, 158

files, 256

Keyword, 305

networks, 325

packagelists, 338

programs, 358

storagespaces, 390

systems, 401

users, 411

verifications, 435

Push Backward, file maintenance function, 261

R

rdb, documenting rdb files, 242
 rdb schema, documenting , 26
 READA, edit field list command, 259
 Recfm, file attribute (ISAM), 252
 Recsize, file attribute (ISAM), 252
 RECSIZE min/max, file attribute (VSAM), 248
 Redefine, Field, 159
 mapping of COBOL data structures, 162
 mapping of Natural data structures, 161
 position and format of redefinition, 160

 Redocumenting, programs, 358
 using default values, 361

 Reflective database, database attribute (Adabas), 19
 Refresh from program, file attribute (Adabas), 195
 Related, file attribute (VSAM), 250
 Related standard file, Field attribute, 120
 Rename
 database, 33
 systems, 401

 Rename File function, 257
 Rename/Change Status of a Verification, 435
 Rename/Move Owner, Owner, 418
 RENUMBER, Rule Editor command, 436
 Replace/append descr., parameter when
 redocumenting programs, 363
 Replacement 1 – 3, verification attribute, 433
 Report Listing
 description of object type, 377
 layout of Report Listing lists, 381

 RESET, editing child lists of programs, 356
 Restrict on drop, file attribute, DB2 table, 222
 Restrictions on input data, documenting Adabas files, 197
 Retrieve, file attribute Entire System Server, 254

Rippling
 concepts of, 265
 impact of changes to Standard Fields, 130
 indicating changes to source Fields, 132

RMS Handler, documenting, 26
 Rotated fields, ESQ, 117
 Rounded up, file attribute (ISAM), 252
 RUN, Rule Editor command, 437

S

S
 suppression option, Field attribute (Adabas C), 118
 variable length option, Field attribute (IMS), 119

 SAVE, Rule Editor command, 436
 Scroll to
 file maintenance parameter, 198
 when defining superfields, 125

 Second default keyword, parameter when
 redocumenting programs, 364
 Secondary
 Field attribute (DB2), 152
 file attribute (ISAM), 252

 Secondary allocation, dataspace attribute, 54
 Security access level, Field attribute (Adabas), 147
 Security update level, Field attribute (Adabas), 147
 Segment name
 file attribute (IMS), 244
 File-Attribut, INGRES Table, 238

 Segment type, file attribute (IMS), 244
 SELECT, editor command, adding Fields to
 Field lists, 207
 Select, File attribute, for SQL views, 204
 Selection Mechanism, when defining derived
 Fields, 125

Predefined Objects in Predict

Sequence field

- File attribute, conceptual files, 200
- file attribute, 189
- file attribute (VSAM), 248, 250
- file attribute Entire System Server, 254

Server, description of the object type, 383

SET ADA, edit field list command, 259

Short name. *See* Field short name

Single-master view, coupling with Master File, rippling, 274

Size for dataspace, dataspace attribute, 56

Size for header, dataspace attribute, 56

Size of RABN, database attribute (Adabas), 20

SORT, documenting Adabas C dataset, 21

SORT [ADA], edit field list command, 259

Sort Flag, Field attribute (VSAM), 139

SORT LOG, editor command, File list of Database, 38

SORT NAME, Extract-specific editor command, 76

SORT PHY, editor command, File list of Database, 38

SORT TYPE, Extract-specific editor command, 76

Sorting Fields and Files, Field retrieval, 167

Source field name

- Field attribute, 127

- Field attribute (DB2), 137

Source-1/2, file attribute (IMS), 244

Source/Xref, program attribute when redocumenting programs, 360

SQL, adding Fields to Field lists of views, 206

SQL clause, generated by rule editor, 433

SQL Files, defining, key or index field, 136

SQL objects, naming conventions, 202

SQL usage. *See* Adabas C SQL usage

Stability, File attribute (Natural Construct), 185

Standard Files, creating, 269

Standard files, documenting, 200

Start

- Field attribute, defining derived Fields, 127

- Field attribute (VSAM), 138

Static SQL, generating DBRMs, 357

Status, verification attribute, 430

Storage clause, File attribute, ORACLE table, 235

Storage group, Field attribute (DB2), 151

Storage pool number, dataspace attribute, 56

Storagegroup name, storagespace attribute, 389

Storagespace, dataspace attribute, 53

Storagespaces

- description of predefined object type, 387

- purging, 390

Sub/superfields/descriptors, determining format, 129

Subfields, specifying length, 128

Subfields/Descriptors, defining, File types A, C and Z, 128

Subpages, Field attribute (DB2), 151

Subquery, edit subquery of SQL view, File maintenance function, 209

subquery, description of (SQL), 209

Superfields, defining, key or index Fields in SQL Files, 136

Superfields/Descriptors, defining, File types A, C and Z, 126

Suppression / null value option, setting

- automatically with profile parameter, 119

Suppression option, Field attribute (Adabas C), 118

SYBASE, naming conventions for SYBASE objects, 237

SYBASE Handler, database type, defining, 30

SYBASE Table, documenting, 238

SYBASE view, documenting, 239

Synchronized, Field attribute (3GL), 142

Synonym of language, Field-specific retrieval parameter, 165

System file (FNAT), database attribute (Adabas), 20

System of type, system attribute, 399

System programs, documenting, 352
 Systems
 description of predefined object type, 397
 layout of system lists, 402
 list of valid types, 399
 purging, 401
 renaming, 401
 specific retrieval parameters, 402

T

Tablespace name
 Dataspace attribute
 DB2, 50
 SQL/DS, 56
 File attribute, ORACLE table, 234
 Target Node, database type, 28
 TEMP, documenting Adabas C dataset, 21
 Type, verification attribute, 432

U

U, unique option, Field attribute, 116
 UISIZE, documenting, 193
 Unique option, Field attribute, 116
 Unit, file attribute (ISAM), 252
 Unused Storagespaces, retrieval function, 391
 UPDATE, editing child list of a program, 356
 Update level, file attribute (Adabas), 195
 Update type, File Relation attribute, 295
 Upgrade Flag, Field attribute (VSAM), 139
 Usage, File Relation attribute (Natural Construct), 295
 User address, 410
 User attributes, 410
 User exit number, Field attribute, hyperdescriptors, 135
 User ISN, file attribute (Adabas), 194

User system DBnr
 program attribute, 346
 system attribute, 400

User system Fnr
 program attribute, 346
 system attribute, 400

Users
 layout of user lists, 412
 purging, 411
 related to no object, maintenance function, 412
 related to objects, maintenance function, 412
 retrieval parameters, 412

Using- and free-block, Field attribute (DB2), 151
 Using/free clause, dataspace attribute, 52

V

Validation program, File attribute, DB2 table, 222
 Value, Field attribute (DB2), 153
 Value format, file attribute (Adabas), 197
 Values, verification attribute, 434
 Variable length option, Field attribute (IMS), 119
 Verifications
 description of predefined object type, 427
 editing rule of a verification, 435
 layout of verification lists, 439
 protecting, 432
 purging, 435
 renaming/changing status, 435
 rule types, 432
 specific retrieval parameters, 438
 valid status, 430
 Verification-specific retrieval parameters, 438

Predefined Objects in Predict

Virtual Machines

- description of predefined object type, 445
- layout of virtual machine lists, 448
- list of valid operating system, 447

Volume, file attribute (VSAM), 248

Volume 1 – 5, file attribute (ISAM), 252

VSAM

- defining primary/alternate superindex, 138
- documenting, 247
 - clusters, 29
- documenting logical files, userviews, 249
- documenting physical VSAM files, 247

VSAM catalog name

- datspace attribute, 53
- Field attribute (DB2), 151
- storagespace attribute, 389

VSAM DD name, File attribute (VSAM), 248

VSAM file org, file attribute (VSAM), 248

VSAM prefix, file attribute (VSAM), 250

W

WHERE clause (SQL), documenting, 210

with Adabas name, Field attribute, function
Push Backward, 262

With check option, File attribute, for SQL
views, 204

WORK, documenting Adabas C dataset, 21

X

Xref data, creating, for object modules, 352

